NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

RECURSIVE PARAMETER IDENTIFICATION FOR ESTIMATING AND DISPLAYING MANEUVERING VESSEL PATH

by

Stephen J. Pollard

December 2003

Thesis Advisor: Roberto Cristi
Co-Advisor: Fotis A. Papoulias

Approved for Public Release; Distribution is Unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY	**2. REPORT DATE** December 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Recursive Parameter Identification for Estimating and Displaying Maneuvering Vessel Path			**5. FUNDING NUMBERS**
6. AUTHOR(S) Pollard, Stephen J.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(maximum 200 words)***

Real–time recursive parameter identification is applied to surface vessel modeling for maneuvering path prediction. An end-to-end system is developed to simulate vessel motion, identify vessel parameters and estimate future path. Path prediction improves bridge team situational awareness by providing a real–time depiction of future motion over the ground on an electronic chart and display system (ECDIS). The extended least–squares (ELS) parameter identification approach allows the system to be installed on most platforms without prior knowledge of system dynamics, provided vessel states are available. The system continually tunes to actual environmental conditions, including vessel ballasting, current, wind and sensor biases. In addition to path prediction, the system estimates maximum vessel roll angle during maneuvering. Maximum roll prediction enhances carrier flight deck safety and increases operational effectiveness by reducing sea room requirements. Suitable performance is demonstrated in real–world maneuvering conditions to recommend that maneuvering path prediction be incorporated into the US Navy's AN/SSN-6 Navigation Sensor System Interface (NAVSSI) electronic charting system. Future research should emphasize an underway demonstration with real–time data acquisition.

14. SUBJECT TERMS recursive parameter estimation, electronic chart and display system, ECDIS, ECN, AN/SSN-6, NAVSSI, vessel dynamic model, path prediction			**15. NUMBER OF PAGES** 175
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

RECURSIVE PARAMETER IDENTIFICATION FOR ESTIMATING AND DISPLAYING MANEUVERING VESSEL PATH

Stephen J. Pollard
Commander, United States Navy
B.S., United States Naval Academy, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2003

Author: Stephen J. Pollard

Approved by: Roberto Cristi
Thesis Advisor

Fotis A. Papoulias
Co-Advisor

John P. Powers
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Real–time recursive parameter identification is applied to surface vessel modeling for maneuvering path prediction. An end-to-end system is developed to simulate vessel motion, identify vessel parameters and estimate future path. Path prediction improves bridge team situational awareness by providing a real–time depiction of future motion over the ground on an electronic chart and display system (ECDIS). The extended least–squares (ELS) parameter identification approach allows the system to be installed on most platforms without prior knowledge of system dynamics, provided vessel states are available. The system continually tunes to actual environmental conditions, including vessel ballasting, current, wind and sensor biases. In addition to path prediction, the system estimates maximum vessel roll angle during maneuvering. Maximum roll prediction enhances carrier flight deck safety and increases operational effectiveness by reducing sea room requirements. Suitable performance is demonstrated in real–world maneuvering conditions to recommend that maneuvering path prediction be incorporated into the US Navy's AN/SSN-6 Navigation Sensor System Interface (NAVSSI) electronic charting system. Future research should emphasize an underway demonstration with real–time data acquisition.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The US Navy is implementing electronic charting and display system (ECDIS) capabilities on naval vessels in an effort to eliminate paper charts and reduce bridge team manpower requirements. Due to unique military requirements, the Navy is procuring the AN/SSN-6 Navigation Sensor System Interface (NAVSSI) vice commercial off the shelf (COTS) technology. Figure 1.1 provides a basic NAVSSI system schematic. The system capabilities include the graphical depiction of a vessel's current state (position and velocity) superimposed on an electronic chart, as well as extensive route planning functionality.

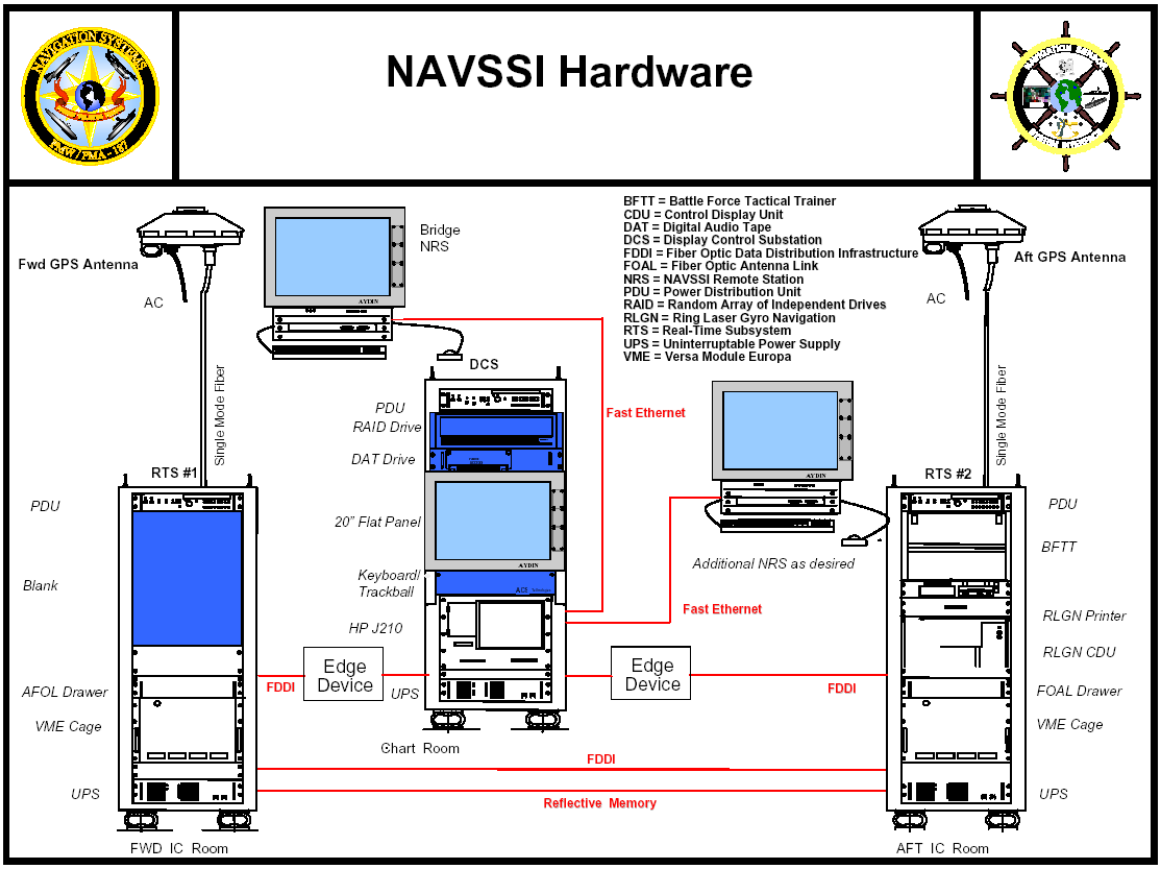


Figure 1.1 AN/SSN-6 Navigation Sensor System Interface (NAVSSI) Diagram

Although a major improvement over paper charting, NAVSSI also serves as a technology inroad for implementing innovative situational awareness enhancements that increase safety and further reduce bridge workload. One enhancement being proposed by the author is the estimation and display of future vessel path during maneuvering situations. Figure 1.2 illustrates the basic concept. Instead of displaying an instantaneous inertial velocity vector, an electronic display should also display estimated vessel path over the ground, taking into account environmental conditions such as vessel loading, wind and current.

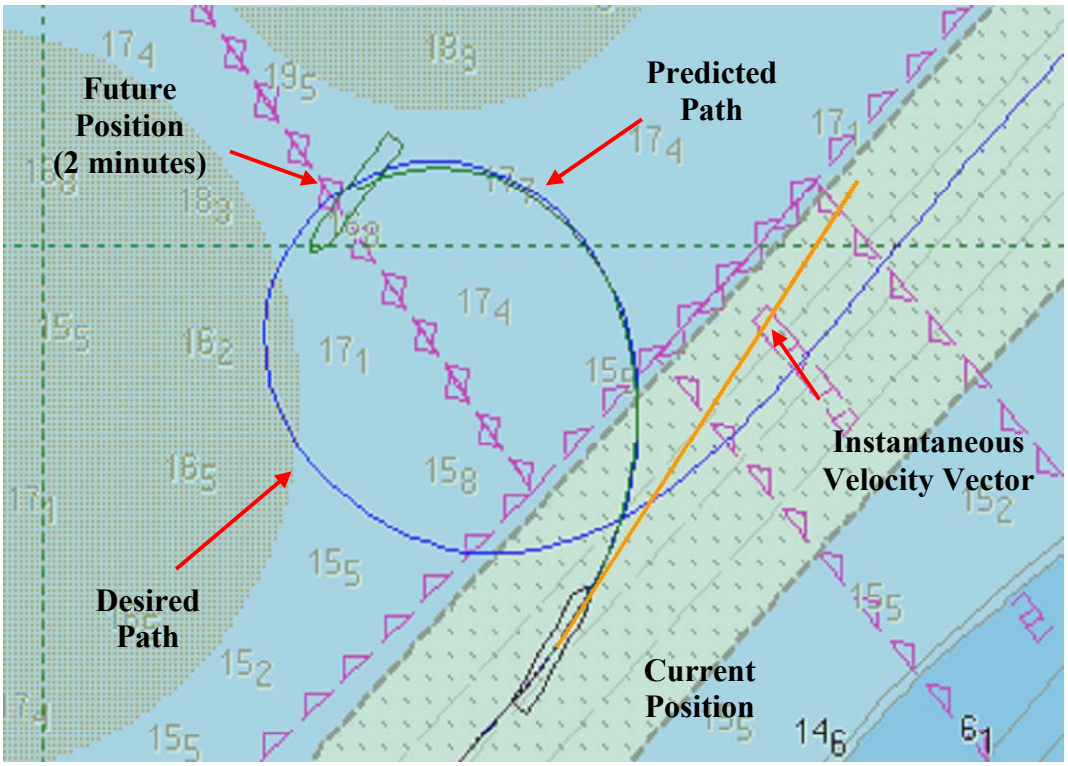


Figure 1.2 Sample Electronic Chart Image Illustrating Difference Between Instantaneous Velocity Vector and Maneuvering Path Prediction

Path prediction enhances situational awareness, especially on large vessels with slow maneuvering time constants (i.e., lag between control input and vessel steady–state response). A single glance at the display allows the Officer of the Deck (OOD) or Commanding Officer to confirm whether current engine and helm orders will effect the desired outcome. Estimated path over the ground proves useful while maneuvering in restricted waters, entering port, negotiating minefields or maneuvering for collision avoid-

ance. In addition, maneuvering path prediction provides for the automatic, real–time depiction of advance and transfer with current and wind taken into account.

From 1998 through 2000, the author developed a path prediction concept demonstrator program called CV Maneuvering Assistant (CV MAST). Figure 1.3 displays a screen snapshot of the CV MAST system simulating an aircraft carrier entering the port of Jebel Ali, Dubai. CV MAST contained a fictitious dynamic model of a four–screw, twin–rudder aircraft carrier that provided sufficient fidelity to serve as a concept demonstration platform. Following favorable comments from bridge watch standers, the author embarked on the current research effort to further refine the concept of path prediction. The primary technical emphasis of this thesis was to develop vessel model structures and parameter identification algorithms that provide a real–time, adaptive estimation of the vessel model and environmental disturbances.

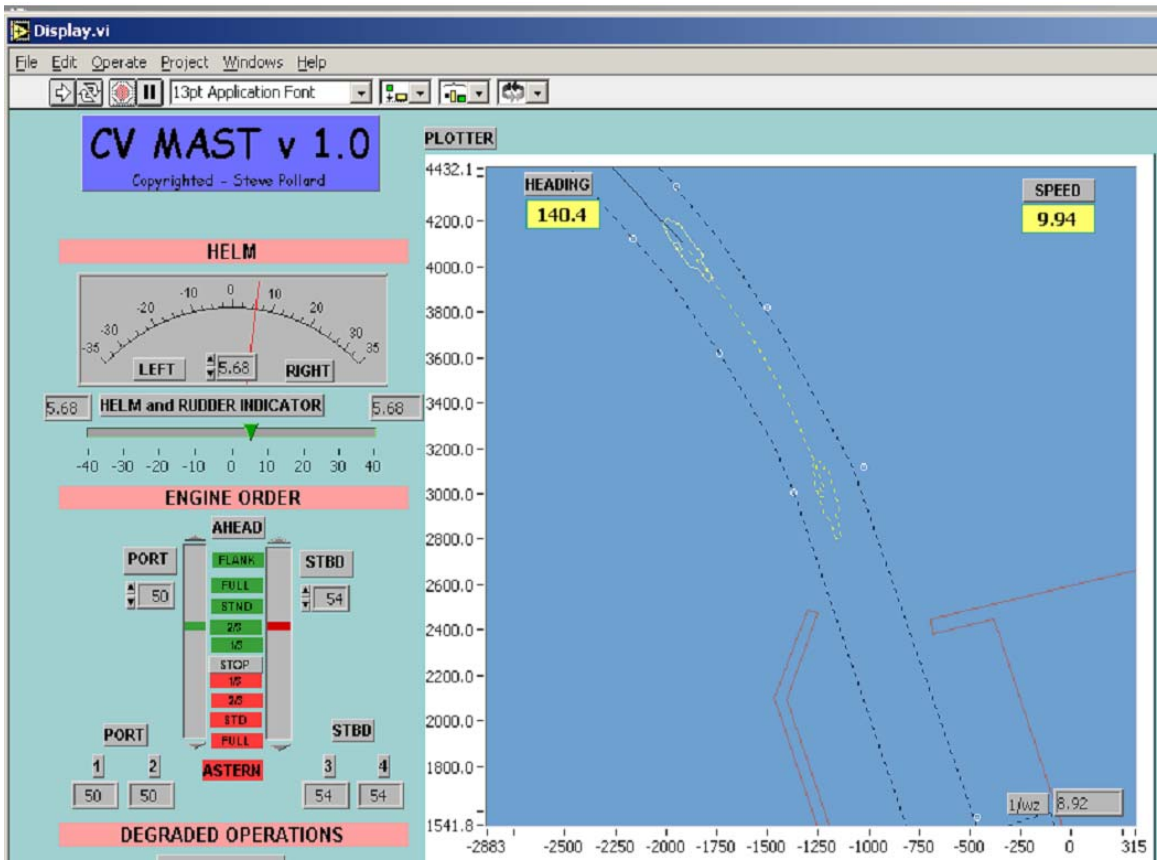


Figure 1.3 CV Maneuvering Assistant Concept Demonstrator

An adaptive model reduces the requirement to create and maintain dynamic models for each class of ship. The approach presented in this research is to have the NAVSSI system identify the dynamic model based on control inputs and observed response. The NAVSSI system is well suited for this function because it already serves as the navigation data fusion center and arbitrates the vessel state. An additional software module would be added to recursively identify vessel model parameters and predict future path taking into account environmental disturbances.

B. STATEMENT OF OBJECTIVES

This research effort had two primary objectives. The first was to demonstrate end-to-end operation of a path prediction system while processing real–world data from a maneuvering vessel. The second primary objective was to investigate vessel model structures and parameter estimation algorithms for automating the task of maintaining an accurate dynamic model. Specific objectives included:

- Identify vessel model structures with sufficient fidelity to provide path prediction over a wide range of operating conditions.
- Develop a stable, parameter estimation algorithm that provides the "core" for the dynamic ship's track estimation. The model should compensate for changes in vessel loading, sensor bias and modeling errors.
- Simulate and demonstrate vessel path prediction overlaid on an electronic chart.
- Investigate operational limitations of the selected path prediction system.
- Investigate whether path prediction has the potential to enhance operational safety and effectiveness.

C. SYSTEM OVERVIEW

A virtual navigation system was designed that simulates vessel motion, estimates parameters, predicts future path and displays the information overlaid on an electronic chart image. LabView by National Instruments (NI) was chosen for the overarching software platform due to its rich graphical user interface and inherent data acquisition

capability. MatLab is used for function block scripting because the author and most technical readers are familiar with MatLab syntax. Figure 1.4 displays the general flow of data through the Data Acquisition, Parameter Identification, Track Prediction and Track Display modules.

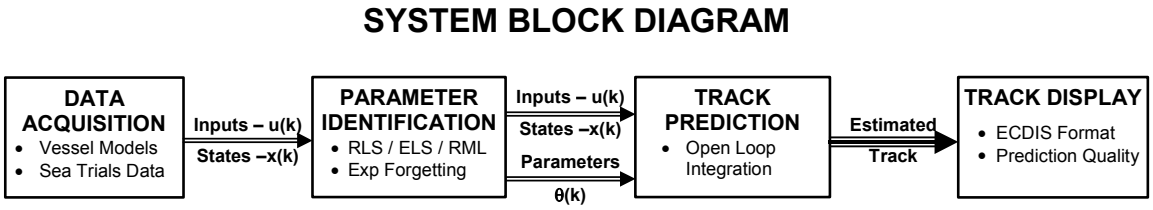


Figure 1.4 Overall System Design Block Diagram

As currently implemented, the system compensates for vessel load, current, wind and rudder sensor bias. Figure 1.5 displays an example screen shot of the system while processing USCGC HEALY sea trials data. The developmental system has many controls and indicators that would not be required on an operational display.

D. THESIS OVERVIEW

Chapter II provides an overview of surface vessel modeling by discussing coordinate systems, rigid body dynamics, dynamic models and non-dimensional parameterization. Chapter III describes the recursive least–squares and extended least–squares parameter identification techniques used to find the model coefficients. Chapter IV describes the overall system design and data flow through the data acquisition, parameter identification, track prediction and track display modules. Results are presented in Chapter V in a build-up approach, culminating with the processing of real–word data from the USCGC Healy Performance and Special Sea Trials. Chapter VI provides conclusions and recommendations, including suggestions for future system enhancements.

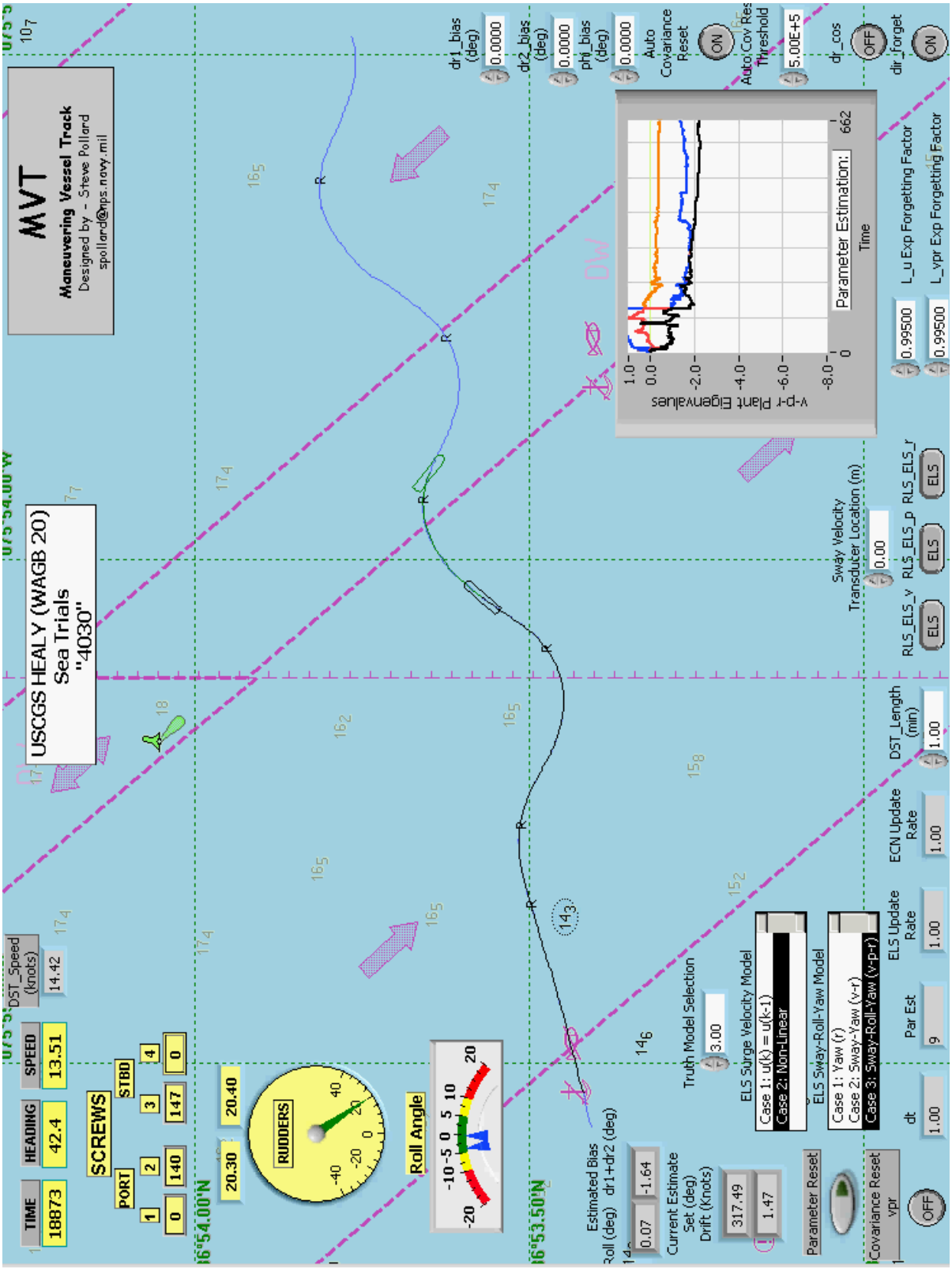


Figure 1.5 Path Prediction System Display

II. SURFACE VESSEL MODELING

Selecting an appropriate dynamic model is critical to achieving desired system performance from both the parameter estimation and path prediction algorithms. Each system block shown in Figure 2.1 relies on vessel modeling. The underlying objective is to select the lowest order model that captures adequate maneuvering fidelity. In this chapter and associated appendices, several dynamic models are presented for the purpose of generating truth data, parameter estimation and track prediction.

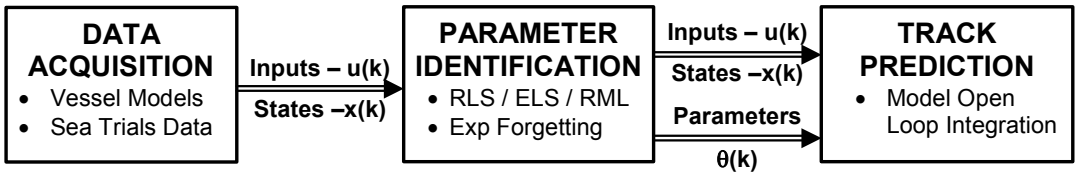


Figure 2.1 System Blocks Requiring Vessel Math Model

Most references present vessel models in continuous–time. However, discrete–time models are more suitable for recursive parameter estimation algorithms. This thesis connects continuous–time models to discrete–time models appropriate for recursive parameter estimation.

A. COORDINATE SYSTEMS

A rigid body surface vessel moves around its center of gravity (CG) relative to an earth–fixed frame of reference. For the slow motion of surface vessels, acceleration due to earth rotation is negligible. Thus, the North–East–Down (NED) tangent plane provides sufficient fidelity for parameter estimation and track prediction. This assumption breaks down close to the north and south poles. However, surface vessel navigation is impracticable at the poles, making the additional overhead of earth–centered, earth–fixed calculations unnecessary. Figure 2.2 displays the body–fixed and NED frames of reference along with the associated velocity and angle notation used in this research. All

frames of reference are right hand systems. A positive rotation of the rudder is in the direction that produces a positive yaw rate while the vessel is moving ahead.

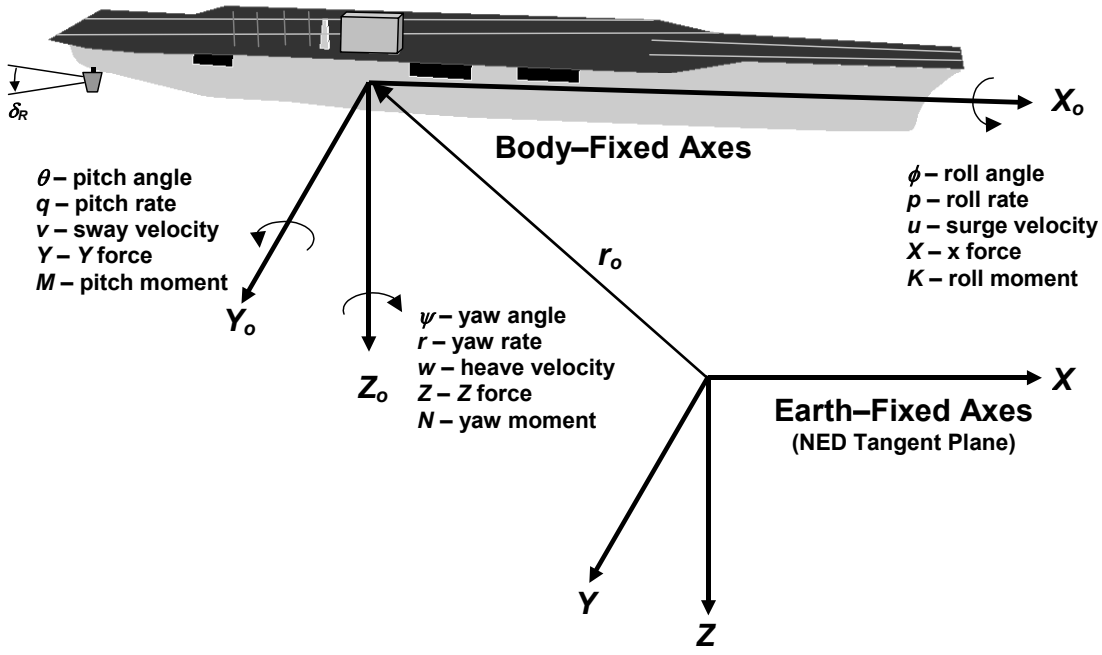


Figure 2.2 Body–Fixed and Earth–Fixed Reference Frames [After Ref. 1.]

B. RIGID–BODY DYNAMICS

All vessel models used in this research are derived from the rigid body equations of motion. Appendix A contains an overview of the six degree of freedom (6–DOF) equations of motion and illustrates how the order may be reduced for vessel parameter estimation and path prediction.

A full 6–DOF model is appropriate in some circumstances, such as modeling the outside visual scene for a bridge simulator. However, simplified models are generally used for surface navigation. In fact, the primary objective of this research is to estimate and display future vessel position on an electronic chart display system. In general, pitch and heave motions can be ignored for surface navigation. By definition, the electronic chart is limited to depicting motion in the *x–y* tangent plane, making a 2–DOF (*u, r*) or 3–

DOF (u ,v, r) model appropriate. However, in some cases it is useful to display roll angle information using a 4–DOF (u ,v, p, r) model. A good example is the estimation and display of maximum roll angle during aircraft carrier maneuvering. This could assist the Officer of the Deck (OOD) with restricting rudder movement during flight operations to prevent aircraft from sliding off the flight deck. Other vessels with roll angle limitations, such as weapon system restrictions, may benefit from roll angle estimation during maneuvering.

The 6–DOF equations of motion are reduced to 4–DOF and simplified with the following assumptions:

1) Vessel is symmetric around the x–z plane ($I_{xy} = I_{yz} = y_G = 0$).

2) Vessel has a homogeneous mass distribution.

3) The body fixed origin is selected as $r_G = \begin{bmatrix} x_G & 0 & z_G \end{bmatrix}^T$ ($I_{xz} = 0$).

4) Pitch and heave are ignored ($q = w = 0$).

The 4–DOF equations of motion are further reduced to 3–DOF by ignoring the roll axis ($p = 0$).

The 3–DOF equations are reduced to 2–DOF by dropping sway motion ($v = 0$).

C. DYNAMIC MODELS

Appendix B provides an overview of linear and non-linear surface vessel models appropriate for recursive parameter estimation. All the models are based on small perturbation decoupling of the sway–yaw–roll modes from the surge equation. The general goal is to find a model with slowly varying parameters that is valid over a large range of operating conditions. Linear models are preferred; however, recursive parameter estimation can handle non-linear models that are linear in parameters [Ref. 2]. For the interested reader, Fossen [Ref. 1] provides an excellent compilation of dynamic models for both surface and sub-surface vessels.

The models presented in Appendix B are extendable to multi–shaft and multi–rudder vessels. However, simply appending additional inputs to the control vector (e.g.,

$\delta_R = \begin{bmatrix} \delta_{R1} & \delta_{R2} \end{bmatrix}^T$) does not provide suitable input for parameter estimation. This is due to lack of independent, persistent excitation of the control inputs. In other words, the rudders and shafts are typically operated in tandem. Heuristically, this means the recursive parameter estimator cannot determine which rudder is providing the control power. The method used in this research is to add control inputs assuming linear superposition as follows:

$$|n|\, n = |n_1|\, n_1 + |n_2|\, n_2 + |n_3|\, n_3 + |n_4|\, n_4 \text{ for multi–shaft vessel} \tag{2.1}$$

$$\delta_R = \delta_{R1} + \delta_{R2} \text{ and } {\delta_R}^2 = {\delta_{R1}}^2 + {\delta_{R2}}^2 \text{ for multi–rudder vessels} \tag{2.2}$$

where n is the shaft speed (RPM) and δ_R is the rudder deflection.

This simplification does create limitations. For instance, summing shaft RPM squared for a multi shaft vessel will not allow the parameter estimator to resolve the "twisting" effect of differential engine orders. However, the nice convergence of control parameters more than offsets limitations during infrequent, non–tandem control inputs. A multiple model approach can be used if additional fidelity is required to take into account twisting situations.

1. Steering Equations

The suitability of three linearized steering (sway–roll–yaw) models is studied for future path prediction. As discussed earlier, sway–roll–yaw motion is decoupled from surge using small perturbation theory. Surge velocity, u, is replaced by mean forward speed, U_0. The following discrete–time models are developed in detail and presented in Appendix B:

a. Nomoto 1st Order Steering Equation – yaw rate only

$$\begin{bmatrix} \psi[k+1] \\ r[k+1] \end{bmatrix} = \begin{bmatrix} 1 & \Delta \\ 0 & f_{22} \end{bmatrix} \begin{bmatrix} \psi[k] \\ r[k] \end{bmatrix} + \begin{bmatrix} 0 \\ g_2 \end{bmatrix} \delta_R[k] \tag{2.3}$$

b. 2nd Order Steering Model – sway and yaw

$$\begin{bmatrix} v[k+1] \\ r[k+1] \\ \psi[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & 0 \\ f_{21} & f_{21} & 0 \\ 0 & \Delta & 1 \end{bmatrix} \begin{bmatrix} v[k] \\ r[k] \\ \psi[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ 0 \end{bmatrix} \delta_R[k] \tag{2.4}$$

c. Coupled Roll–Sway–Yaw Steering Model

$$\begin{bmatrix} v[k+1] \\ r[k+1] \\ p[k+1] \\ \phi[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ 0 & 0 & \Delta & 1 \end{bmatrix} \begin{bmatrix} v[k] \\ r[k] \\ p[k] \\ \phi[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ 0 \end{bmatrix} \delta_R[k] \tag{2.5}$$

2. The Speed Equation

Two surge models are investigated. The most basic model assumes constant forward speed ($u[k] = u[k-1]$). This turns out to be an adequate assumption most of the time, due to the typical large surge velocity time constants of ships and the relatively short path prediction interval. However, high bandwidth maneuvering situations require more model fidelity. Linear surge models are available, but they are only valid for small ranges around the trimmed vessel speed. As a result, multiple linear trim points are required to model a typical vessel speed range. Assuming *u, v* and *r* are observable vessel states; it is relatively straightforward to develop a non-linear speed equation that is linear in parameters. This is the second model investigated:

$$\begin{aligned} u[k+1] = f_1\, u[k] + f_2\, |u[k]|\, u[k] + f_3\, v[k]\, r[k] \ldots \\ + f_4\, r[k]^2 + g_1\, \delta_R[k]^2 + g_2\, |n[k]|\, n[k] \end{aligned} \tag{2.6}$$

where:

$$\text{Thrust} \propto n^2 \text{ and } n \cong \text{Trim RPM} \tag{2.7}$$

$$|n|\, n = |n_1|\, n_1 + |n_2|\, n_2 + |n_3|\, n_3 + |n_4|\, n_4 \text{ for multi shaft vessels} \tag{2.8}$$

$$\delta_R{}^2 = \delta_{R1}{}^2 + \delta_{R2}{}^2 \text{ for multi rudder vessels.} \tag{2.9}$$

D. NON-DIMENSIONAL PARAMETERIZATION

It is useful to non-dimensionalize model parameters with respect to vessel speed. This keeps parameter coefficients stable over a wider range of operating conditions. Several systems of normalization exist, but the most common is the Prime system. Appendix C displays the normalization parameters for the Prime – I (′) and Prime – II (″) systems. For consistency, all normalization in this paper will use the Prime – I system. The typical vessel states for a 4–DOF model are non-dimensional in the Prime – I system as follows:

$$U = \sqrt{u^2 + v^2}\ ;\ u = U\,u';\quad v = U\,v';\quad p = \frac{U}{L}\,p';\quad r = \frac{U}{L}\,r';\quad \delta_R = \delta_R{}' \qquad (2.10)$$

E. DISCRETE SYSTEM INTEGRATION TECHNIQUES

Data is generated from vessel math models by discrete–time integration of continuous systems. Several techniques are available that range in fidelity and computational cost. Appendix E summarizes three popular algorithms and the corresponding truncation error. The Data Acquisition module uses the Runge–Kutta 2nd Order algorithm to generate simulated vessel data. Note that surface vessels typically display light damping in the roll mode. Any open-loop integration of models with a roll mode should use the Runge–Kutta 2nd Order or higher techniques.

F. CHAPTER SUMMARY

Chapter II provided an overview of vessel modeling. The body axis frame of reference was introduced relative to the North–East–Down inertial plane. Simplified dynamic models were developed from rigid body dynamics. The Prime – I System was used to non-dimensionalize model parameters. Chapter III will present the recursive parameter estimation techniques for on–line parameter identification routines. Both the recursive and extended least–squares algorithms are discussed. In addition, several methods are presented for coping with time–varying model parameters, including exponential forgetting and conditional updating.

III. PARAMETER IDENTIFICATION/ESTIMATION

Most texts provide extensive examples of least–squares for SISO systems, but leave it to the reader to implement the algorithm for MIMO systems. This chapter illustrates how to implement the RLS and ELS algorithms for MIMO systems typical of vessel models.

The Parameter Identification module implements the recursive least–squares (RLS) and extended least–squares (ELS) algorithms. Ljung [Ref. 3] is an excellent text covering both the mathematical derivation and historical background of recursive parameter estimation techniques. Ljung goes to great effort to point out that least–squares is a subset of similar algorithms with slightly different assumptions. In this case, least–squares techniques were selected due to widespread use and a straightforward implementation.

For least–squares techniques to work, one must select a linear or linear in parameters model with sufficient fidelity to capture the desired systems dynamics. Vessel models that capture varying motion complexity were presented in Chapter II. Specific implementation in the Parameter Identification module is discussed in Chapter IV.

A. LEAST–SQUARES ESTIMATION

This thesis is based on parameter identification using various realizations of the least–squares technique. The general least–squares formulation is traced to Gauss (1809) [Ref. 3]. Least–squares states that the unknown parameters of a mathematical model should be chosen in such a way that "the sum of the squares of the differences between actually observed and computed values, multiplied by numbers that measure the degree of precision, is a minimum" [Ref. 1].

Least–squares can be applied to models of the form:

$$y(i) = \phi_1(i)\,\theta_1 + \phi_2(i)\,\theta_2 \cdots + \phi_n(i)\,\theta_n = \phi^T(i)\,\theta \tag{3.1}$$

where $y(i)$ is the observed value, θ is a vector of parameters of the model to be determined and $\phi^T(i)$ is a vector of regression variables. The parameters of the model, θ, are chosen to minimize the least–squares loss function $V(\theta,t)$:

$$V(\theta,t)=\frac{1}{2}\sum_{i=1}^{t}\left(y(i)-\phi^T(i)\,\theta\right)^2. \tag{3.2}$$

Provided y is linear in the model parameters (θ), the least–squares criterion is quadratic with an analytic solution. The cost function is minimized by setting the gradient to zero or by completing the square and solving for the minimum. For either method, the analytic solution is the "normal equation:"

$$\theta(t)=\hat{\theta}(t)=\left(\sum_{i=1}^{t}\phi(i)\,\phi^T(i)\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)\,y(i)\right)=P(t)\left(\sum_{i=1}^{t}\phi(i)\,y(i)\right) \tag{3.3}$$

where

$$P(t)=\left(\sum_{i=1}^{t}\phi(i)\,\phi^T(i)\right)^{-1}. \tag{3.4}$$

A statistical interpretation of least–squares is that the model parameter estimates ($\hat{\theta}$) approach the true model parameters (θ) as the number of observations approach infinity, provided the process is in the form of:

$$y(i)=\phi^T(i)\,\theta+e(i) \tag{3.5}$$

where θ is a vector of true model parameters, and $e(i)$ is a vector of independent, identically distributed, zero mean random variables. Extended least–squares (ELS) extends the concept to colored noise of the process $v(k)=C\left(z^{-1}\right)e(k)$, where C contains additional parameters to estimate.

B. RECURSIVE LEAST–SQUARES

The general least–squares technique is not suited for real–time, on–line parameter estimation due to the requirement to batch process large data sets. As a result, many au-

thors independently developed a recursive process, although the earliest reference is attributed to Plackett (1950) [Ref. 3].

1. Difference Equation Models

The starting point for recursive least–squares is a difference equation model in the form of:

$$y(k)+a_1 y(k-1)+\ldots+a_n y(k-n)=b_1 u(k-1)+\ldots+b_m y(k-m)+v(k) \tag{3.6}$$

where $v(t)$ is an unspecified noise process. Using z^{-1} as the backward time shift operator, the equation may be written compactly as:

$$A\left(z^{-1}\right) y(k)=B\left(z^{-1}\right) u(k)+v(k) \tag{3.7}$$

where *A* and *B* are polynomials of the delay operator z^{-1}:

$$A\left(z^{-1}\right)=1+a_1 z^{-1}+a_2 z^{-2}+\ldots+a_n z^{-n} \tag{3.8}$$

$$B\left(z^{-1}\right)=b_1 z^{-1}+b_2 z^{-2}+\ldots+b_m z^{-m}. \tag{3.9}$$

For recursive methods, the system is reorganized into linear regression form using a model parameter vector (θ), a lagged input-output data vector (ϕ) and a noise process (v):

$$y(k)=\phi^T(k)\theta+v(k) \tag{3.10}$$

where

$$\theta=\left[a_1,\ldots,a_n,\ b_1,\ldots,b_m\right] \tag{3.11}$$

$$\phi(k)=\left[-y(k-1),\ \ldots,-y(k-n),\ u(k-1),u(k-2),\ldots,u(k-m)\right]^T. \tag{3.12}$$

The components of the data vector (ϕ) are referred to as regression variables.

2. Recursive Computations

The general least–squares algorithm may be solved using recursive equations. The results from the last iteration, (k – 1), are used to obtain an estimate of system parameters at the current time (k) without reprocessing the entire realization. Essentially, all previous data is processed recursively and stored in the parameter estimates, regres-

sion variables and covariance matrix. A new parameter estimate is obtained by adding a correction proportional to the error between the measured value and the predicted value using the last parameter estimate:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\varepsilon(k) \tag{3.13}$$

where

$$\varepsilon(k) = y(k) - \varphi^T(k)\,\hat{\theta}(k-1). \tag{3.14}$$

3. The ARX Model and Recursive Least–Squares

As of yet, no attempt has been made to characterize the disturbance term (v). When the noise process is characterized as a white noise sequence, $e(t)$, the model becomes:

$$A(z^{-1})y(k) = B(z^{-1})u(k) + e(k). \tag{3.15}$$

With parameter and data vectors:

$$\theta = [a_1, \ldots, a_n, b_1, \ldots, b_m] \tag{3.16}$$

$$\varphi(k) = [-y(k-1),\ \ldots, -y(k-n),\ u(k-1),\ \ldots, u(k-m)]^T. \tag{3.17}$$

This model is commonly referred to as the ARX model. It is a combination of an autoregressive part (AR) $A(z^{-1})\,y(k)$ and a control term $B(z^{-1})\,u(k)$. The control portion is termed an exogenous variable (X) from economic literature [Ref. 3]. If the disturbance is other than white noise, the ARX model will provide biased estimates of the parameters.

ARX model parameters are estimated using the recursive least–squares algorithm. The RLS equations are listed here without proof:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\left(y(k) - \varphi^T(k)\,\hat{\theta}(k-1)\right) \tag{3.18}$$

$$K(t) = P(t)\varphi(t) = P(t-1)\varphi(t)\left(I + \varphi^T(t)\,P(t-1)\varphi(t)\right)^{-1} \tag{3.19}$$

$$\begin{aligned} P(t) &= P(t-1) - P(t-1)\varphi(t)\left(I + \varphi^T(t)\,P(t-1)\varphi(t)\right)^{-1} \\ &= \left(I - K(t)\varphi^T(t)\right)P(t-1) \end{aligned} \tag{3.20}$$

where $\hat{\theta}(t)$ is a vector of model parameter estimates, $K(t)$ is the weighting factor vector for combining the correction with the previous estimate and $P(t)$ is the covariance matrix.

The derivation and proof of the RLS equations is beyond the scope of this thesis. For the interested reader, Ljung and Södersröm [Ref. 4] and Astrom and Wittenmark [Ref. 2] are straightforward texts covering the subject.

The RLS parameter estimate can be thought of as a Kalman filter for the process [Ref. 2]:

$$\theta(t+1) = \theta(t) \tag{3.21}$$

$$y(t) = \varphi^T(t)\theta(t) + e(t). \tag{3.22}$$

Many real–world systems are driven by colored noise disturbances. Surface vessels are disturbed by many factors including sea state. Wave action is a colored noise process, which may be approximated by a first–order wave model. If RLS is used, the signal should be filtered to remove or reduce colored noise prior to parameter estimation.

4. Recursive Least–Squares and MIMO Systems

The RLS algorithm may be extended to multi–input, multi–output (MIMO) systems. The standard RLS equations are used. But, the regression model is augmented to appropriate dimensions:

$$y(k) = \Phi^T(k)\,\theta + e(k). \tag{3.23}$$

An example is illustrated with the linearized, sway–yaw model. For MIMO systems it is easiest to start from a discrete state–space representation of the system:

$$x[k] = F\,x[k-1] + G\,u[k-1] + v[k]. \tag{3.24}$$

Assuming white noise disturbance, the sway–yaw model expands to:

$$\begin{bmatrix} v[k] \\ r[k] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} v[k-1] \\ r[k-1] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \delta_R[k-1] + \begin{bmatrix} e_1[k] \\ e_2[k] \end{bmatrix}. \tag{3.25}$$

The system is re-arranged in linear regression format:

$$y[k] = \varphi^T[k]\,\theta[k-1] + \begin{bmatrix} e_1[k] & e_2[k] \end{bmatrix} \tag{3.26}$$

where:

$$y[k] = \begin{bmatrix} v[k] & r[k] \end{bmatrix} \tag{3.27}$$

$$\varphi^T[k] = \begin{bmatrix} v[k-1] & r[k-1] & \delta_R[k-1] \end{bmatrix} \tag{3.28}$$

$$\theta = \begin{bmatrix} F^T \\ G^T \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ g_1 & g_2 \end{bmatrix}. \tag{3.29}$$

Note that the RLS equations can be compactly implemented in matrix form without breaking the algorithm into a system of SISO equations. This is because the covariance term, $P(k)$, is solely a function of the regression vector, $\varphi(k)$. In the sway–yaw model, the regression vector is identical for both the sway and yaw models. When the RLS algorithm is extended to colored noise processes (ELS), this no longer holds true and the RLS equations must be treated as a system of multi–input, single–output models.

5. The ARMAX Model and Extended Least–Squares

Parameters may be estimated for systems with colored noise disturbances using the ARMAX model. The disturbance process is characterized as a moving average of the white noise sequence $e(t)$:

$$v(k) = C\left(z^{-1}\right) e(k) \tag{3.30}$$

$$C\left(z^{-1}\right) = 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_r z^{-r} \tag{3.31}$$

The model becomes:

$$A\left(z^{-1}\right) y(k) = B\left(z^{-1}\right) u(k) + C\left(z^{-1}\right) e(k) \tag{3.32}$$

with parameter and data vectors:

$$\theta = \left[a_1, \ldots, a_n, b_1, \ldots, b_m, c_1, \ldots, c_r\right] \tag{3.33}$$

$$\begin{aligned}\varphi(k) = \big[&-y(k-1),\ \ldots, -y(k-n),\ u(k-1),\\ &\ldots, u(k-m),\ e(k-1),\ \ldots, e(k-r)\big]^T.\end{aligned} \tag{3.34}$$

This model is commonly referred to as the ARMAX model. It is a combination of an autoregressive part (AR) $A\left(z^{-1}\right)y(k)$, a moving average part (MA) $C\left(z^{-1}\right)e(k)$ and a control part $B\left(z^{-1}\right)u(k)$. The control portion is termed an exogenous variable (X) from economic literature [Ref. 3]. The ARMAX model can deal with first–order wave disturbances directly without pre-filtering the data.

The recursive algorithm for the ARMAX model is termed extended least–squares (ELS). The algorithm uses the RLS equations with augmented data and parameter vectors. The error parameter, $e(k)$, is not known, but may be approximated by the prediction error:

$$\varepsilon(k) = y(k) - \varphi^T(k)\,\hat{\theta}(k-1) \tag{3.35}$$

with

$$\hat{\theta} = \left[\hat{a}_1, \ldots, \hat{a}_n, \hat{b}_1, \ldots, \hat{b}_m, \hat{c}_1, \ldots, \hat{c}_r\right] \tag{3.36}$$

$$\begin{aligned}\varphi(k) = \big[&-y(k-1),\ \ldots, -y(k-n),\ u(k-1),\\ &\ldots, u(k-m),\ \varepsilon(k-1),\ \ldots, \varepsilon(k-r)\big]^T.\end{aligned} \tag{3.37}$$

In most cases it is advantageous to update and store the prediction error based on the latest estimate of the model parameters instead of the $(k-1)$ update. This is termed the posterior update:

$$\varepsilon(k) = y(k) - \varphi^T(k)\,\hat{\theta}(k). \tag{3.38}$$

6. Extended Least–Squares and MIMO Systems

ELS is best applied to MIMO systems by breaking the model up into a system of multi–input, single–output regression equations. The individual models are solved using

the ELS algorithm and then reassembled into a MIMO system. The sway–yaw model is used as an example:

$$x[k] = F\,x[k-1] + G\,u[k-1] + v[k]. \tag{3.39}$$

RLS assumed the disturbance is white noise of the form $v[k] = C(z^{-1})\,e[k]$, where $C(z^{-1}) = 1$. ELS adds the flexibility to process colored noise disturbances of the general form $v[k] = C(z^{-1})\,e[k]$. Assuming colored noise disturbances, the sway–yaw model expands to:

$$\begin{bmatrix} v[k] \\ r[k] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} v[k-1] \\ r[k-1] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \delta_R[k-1] + \begin{bmatrix} C_1(z^{-1})\,e_1[k] \\ C_2(z^{-1})\,e_2[k] \end{bmatrix}. \tag{3.40}$$

The model is arranged as a system of equations in linear regression format:

$$\begin{bmatrix} y_1[k] \\ y_2[k] \end{bmatrix} = \begin{bmatrix} \varphi_1^{\,T}[k]\,\theta_1[k-1] \\ \varphi_2^{\,T}[k]\,\theta_2[k-1] \end{bmatrix} \tag{3.41}$$

where:

$$\begin{bmatrix} y_1[k] \\ y_2[k] \end{bmatrix} = \begin{bmatrix} v[k] \\ r[k] \end{bmatrix} \tag{3.42}$$

$$\varphi_1^{\,T}[k] = \begin{bmatrix} v[k-1] & r[k-1] & \delta_R[k-1] & \varepsilon_v[k-1] \ldots \varepsilon_v[k-r] \end{bmatrix} \tag{3.43}$$

$$\varphi_2^{\,T}[k] = \begin{bmatrix} v[k-1] & r[k-1] & \delta_R[k-1] & \varepsilon_r[k-1] \ldots \varepsilon_r[k-r] \end{bmatrix} \tag{3.44}$$

$$\theta_1 = \begin{bmatrix} f_{11} & f_{12} & g_1 & c_{11} & \ldots & c_{1r} \end{bmatrix}^T \tag{3.45}$$

$$\theta_2 = \begin{bmatrix} f_{21} & f_{22} & g_2 & c_{21} & \ldots & c_{2r} \end{bmatrix}^T \tag{3.46}$$

$$\theta = \begin{bmatrix} F^T \\ G^T \\ C^T \end{bmatrix} = \begin{bmatrix} f_{11} & f_{21} \\ f_{12} & f_{22} \\ g_1 & g_2 \\ c_{11} & c_{21} \\ \vdots & \vdots \\ c_{1r} & c_{2r} \end{bmatrix}. \tag{3.47}$$

The covariance matrix must be solved for each individual multi–input, single–output model, resulting in increased computational and memory requirements compared to the RLS matrix MIMO solution. However, there is one benefit to this approach. The covariance matrix now applies individually to the output state. In this case, the sway and yaw rate errors are tracked separately and one obtains separate covariance matrices for each. This proves useful for estimating the bounds of the observed error for each state.

7. The ARMAX Model and Recursive Maximum Likelihood

Recursive maximum likelihood (RML) is a modification to the ELS algorithm. The regression vector (φ) is replaced by a regression vector filtered by the noise parameter coefficients, $C\left(z^{-1}\right)$ in the gradient portion of the RLS equations. The filtered regression vector (ψ) becomes [Ref. 1]:

$$C\left(z^{-1}\right)\psi(k) = \varphi(k) \quad \Rightarrow \quad \psi(k) = \frac{\varphi(k)}{C\left(z^{-1}\right)}. \tag{3.48}$$

The RML equations are:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\left(y(k) - \varphi^T(k)\,\hat{\theta}(k-1)\right) \tag{3.49}$$

$$K(t) = P(t)\,\psi(t) = P(t-1)\,\psi(t)\left(I + \psi^T(t)\,P(t-1)\,\psi(t)\right)^{-1} \tag{3.50}$$

$$P(t) = \left(I - K(t)\,\psi^T(t)\right)P(t-1). \tag{3.51}$$

8. Nonlinear Models

The recursive algorithms may be used for nonlinear systems provided the model is linear in parameters. The nonlinear surge equation is employed as an example. From Chapter II, the simplified surge equation in discrete–time is given by:

$$\begin{aligned} u[k+1] = f_1\, u[k] + f_2 \left|u[k]\right| u[k] + f_3\, v[k]\, r[k] \ldots \\ + f_4\, r[k]^2 + g_1\, \delta_R[k]^2 + g_2 \left|n[k]\right| n[k]. \end{aligned} \tag{3.52}$$

This nonlinear model is considered linear in parameters when the regression vector and parameters are defined as follows:

$$\begin{aligned} \varphi(k) = \Big[u[k-1], \quad \left|u[k-1]\right| u[k-1], \quad v[k-1]\; r[k-1], \\ \ldots r[k-1]^2, \quad \delta_R[k-1]^2, \quad \left|n[k-1]\right| n[k-1] \Big]^T \end{aligned} \tag{3.53}$$

$$\theta = \left[f_1, f_2, f_3, f_4, g_1, g_2 \right]^T. \tag{3.54}$$

C. TIME–VARYING PARAMETERS

The recursive parameter estimation algorithms discussed in Section B were developed from time–invariant system models. However, in practice, most systems contain some degree of time–varying parameters. Surface vessel models definitely contain slowly varying parameters. For instance, vessel ballasting will change due to fuel burn on a long voyage, resulting in a slow variation of the mass and center of gravity sensitive parameters. In addition, linearized models are inherently time–varying due to the nonlinearity in the underlying system. Maneuvering a vessel outside the valid range of a linear model (i.e., large speed change or hard rudder command) will require a time–varying algorithm to capture the abrupt shift in model parameters. Two extensions to the standard recursive algorithms, exponential forgetting and parameter resetting, are typically used to capture time–varying model parameters.

1. Exponential Forgetting

Exponential forgetting is a method to discount "older" samples of data so the most recent samples receive the greatest weighting. It is well suited for slowly varying

parameters. To obtain exponential forgetting, the least–squares criteria is adjusted such that the samples that are 'n' units old are weighted by the factor λ^n [Ref. 1]:

$$V(\theta,t)=\frac{1}{2}\sum_{i=1}^{t}\lambda^{t-i}\left(y(i)-\phi^T(i)\,\theta\right)^2. \tag{3.55}$$

The recursive algorithm for RLS, ELS and RML with exponential forgetting, becomes:

$$\hat{\theta}(k)=\hat{\theta}(k-1)+K(k)\left(y(k)-\varphi^T(k)\,\hat{\theta}(k-1)\right) \tag{3.56}$$

$$K(t)=P(t)\,\varphi(t)=P(t-1)\,\varphi(t)\left(\lambda I+\varphi^T(t)\,P(t-1)\,\varphi(t)\right)^{-1} \tag{3.57}$$

$$\begin{aligned}P(t)&=P(t-1)-P(t-1)\,\varphi(t)\left(I+\varphi^T(t)\,P(t-1)\,\varphi(t)\right)^{-1}\\&=\left(I-K(t)\,\varphi^T(t)\right)P(t-1)/\lambda\end{aligned} \tag{3.58}$$

where $\hat{\theta}(t)$ is a vector of model parameter estimates, $K(t)$ is the weighting factor vector for combining the correction with the previous estimate, $P(t)$ is the covariance matrix and λ is the forgetting factor ($0<\lambda\le 1$). Note that setting $\lambda=1$ will revert the algorithm to the standard recursive equations.

The exponential forgetting algorithm is easily implemented. However, care must be taken to prevent covariance "wind–up." Under no excitation, the covariance matrix will grow exponentially. One method to deal with wind–up is to switch on forgetting only during periods of persistent excitation. This is termed "conditional forgetting." One method is to update the parameter estimate and covariance if [Ref. 1]:

$$\varphi[k]^T\,P(k)\,\varphi[k]>2(1-\lambda). \tag{3.59}$$

Exponential forgetting may also be used to discount the initial start-up transient. After a suitable number of samples are processed and the parameter estimates are settling down, a high forgetting factor (small λ) may be switched on for a brief period. This will discount the initial samples and allow for further refinement of the parameter estimates.

An exponential forgetting time constant is selected to correspond with the rate of change in the model parameters. The forgetting time constant is obtained from:

$$\lambda = e^{-\Delta/T_f} \tag{3.60}$$

where T_f is the exponential forgetting time constant, λ is the forgetting factor and Δ is the sample interval.

An appropriate forgetting factor may also be selected from Table 3.1. One must keep in mind that a shorter time constant results in more noise corruption of the parameter estimate. This is due to the reduction in smoothing obtained from larger sample sets.

T_f/Δ	λ
1	0.37
2	0.61
5	0.82
10	0.90
20	0.95
50	0.98
100	0.99

Table 3.1 Exponential Forgetting Factor Time Constant [From Ref. 1.]

2. Covariance Resetting

Covariance resetting is a method to deal with abrupt changes in model parameters. The basic concept is to reset the covariance matrix, *P*, to a fixed constant times the identity matrix, $P = \alpha I$. Resetting is commanded periodically or when specific criteria are met based on a priori knowledge of the system time variation. Covariance resetting is highly effective when commanded at appropriate times, but will introduce noise in the parameter estimate if commanded during a period of static model parameters [Ref. 1].

3. Conditional Updating

One drawback to exponential forgetting is covariance wind–up. During periods of low excitation, the covariance will increase exponentially. During the next period of excitation, covariance wind–up will have increased the gain to the point of causing erratic parameter estimates.

Conditional updating is one method to mitigate windup. The general concept is similar to conditional forgetting. The estimate and covariance matrix are only updated when excitation exceeds a specified level. If too stringent a condition is picked, the parameter estimates will be updated infrequently. If too small a criterion is selected, covariance wind–up will occur. A suggested method is to use the test case [Ref 1]:

$$\varphi(t)^T P(t)\,\varphi(t) > 2(1-\lambda). \tag{3.61}$$

4. Directional Forgetting

Another drawback of standard exponential forgetting is the inability to control forgetting in multiple input systems. All regression variables are treated the same way. Insufficient excitation in one or more regression variables will cause covariance windup in that term. Even conditional updating won't prevent covariance windup. A solution is to implement forgetting in the direction of the excitation. There are many variations of directional forgetting. Astrom [Ref 1] suggests using the following method to update the covariance matrix in the direction of excitation:

$$P^{-1}(t+1) = P^{-1}(t) + \left(1 + (\lambda - 1)\frac{\varphi^T(t)\,P^{-1}(t)\,\varphi(t)}{\left(\varphi^T(t)\;\varphi(t)\right)^2}\right)\varphi(t)\,\varphi^T(t). \tag{3.62}$$

D. CHAPTER SUMMARY

This chapter discussed recursive parameter estimation techniques for identifying model parameters. Both the recursive and extended least–squares algorithms were presented. Exponential forgetting, covariance resetting, conditional updating and directional forgetting are techniques for dealing with time–varying parameters. Chapter IV will step through the overall system design. Actual model parameterization structures are covered in detail.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DYNAMIC VESSEL TRACK SYSTEM DESIGN

As discussed in the introduction, LabView by National Instruments (NI) was chosen for the overarching software platform due to a rich, built-in graphical user interface and inherent data acquisition capability. LabView function block scripting can be performed with the native HiQ language or with MatLab through an X-Windows interface. The MatLab approach was chosen because the author and most readers are familiar with MatLab syntax. This chapter describes the design features implemented in the Data Acquisition, Parameter Identification, Track Prediction and Track Display modules.

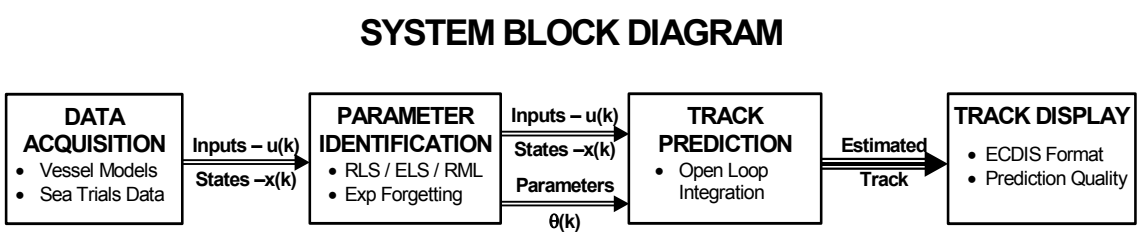


Figure 4.1 Overall System Block Diagram

A. DATA ACQUISITION MODULE

The Data Acquisition Module generates truth data from various vessel math models or processes sea trials data. In the math model mode, the user interactively operates the vessel controls or loads a predetermined sequence of control inputs. In the sea trials mode, control input and vessel states are obtained from stored sea trial data files. Future enhancements could include a real–time data capture capability.

The Data Acquisition module outputs the 23–term state vector, including control inputs, in the following format:

$$x[t] = \left[u\ v\ p\ r\ \phi\ \psi\ x_{pos}\ y_{pos}\ \delta_{R1}\ \delta_{R2}\ n_1\ n_2\ n_3\ n_4\ t\ x_{lat}\ y_{long}\ V_{RW}\ \psi_{RW} \ldots \right.$$
$$\left. \ldots\ V_{Ci\_x}\ V_{Ci\_y}\ V_{i\_x}\ V_{i\_y} \right] \tag{4.1}$$

where:

u – Surge Velocity (m/s) v – Sway Velocity (m/s)

p – Roll Rate (rad/s) r – Yaw Rate (rad/s)

ϕ – Roll Angle (rad) ψ – Yaw Angle/True Heading (rad)

x_{pos} – Relative Position (North-m) y_{pos} – Relative Position (East-m)

δ_{R1}/δ_{R1} – Rudder Angles (rad) $n_1/n_2/n_3/n_4$ – Shaft Turns (RPM)

t – Time (sec)

x_{lat} – Latitude (decimal deg) y_{long} – Longitude (decimal deg)

V_{RW} – Relative Wind (m/s) ψ_{RW} – Relative Wind Direction (rad)

V_{Ci-x} – Current in x-direction (m/s) V_{Ci-y} – Current in y-direction (m/s)

V_{i-x} – Inertial Velocity (x - m/s) V_{i-y} – Inertial Velocity (y - m/s).

Note: Hooks are in place for relative wind direction and speed. Wind is not currently modeled. Wind effects are captured in the general disturbance factor. Growth options could include wind parameter estimation.

1. Vessel Math Model Mode

The Math Model Mode generates vessel state data from a known math model for validation of the Parameter Identification module. Several model structures are selectable, all of which are based on the "Containership" model developed by Fossen [Ref. 1]. This model was selected due to its widespread use in vessel dynamics literature. Table 4.1 summarizes the model structure cases. All lateral–directional model parameters are non-dimensional using the Prime I (′) system.

CASE		MODEL STRUCTURE	WAVE MODEL
1	Surge:	$\dot{u} = 0, \quad U_0 = 7\ m/s$	
	Linear Yaw - Nomoto's 1st Order	$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta_R$	None
2	Surge:	$\dot{u} = 0, \quad U_0 = 7\ m/s$	
	Linearized, Coupled Sway-Yaw	$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta_R$	None
3	Surge:	$\dot{u} = 0, \quad U_0 = 7\ m/s$	
	Linearized, Coupled Sway-Roll-Yaw	$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} & a_{15} \\ a_{21} & a_{22} & 0 & a_{24} & a_{25} \\ 0 & 1 & 0 & 0 & 0 \\ a_{41} & a_{42} & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ b_4 \\ 0 \end{bmatrix} \delta_R$	1st Order Roll and Yaw Random Wave Models (See Chapter II, Section C)
4	3rd Order N-L Surge-Sway-Roll-Yaw	Coupled 4-DOF N-L Equations of Motion with 1st Order Roll–Yaw Wave Model	1st Order Roll and Yaw Random Wave Models (See Chapter II, Section C)

Table 4.1 Data Acquisition Module Math Models

2. Sea Trials Data Mode

The Sea Trials Data Mode extracts raw sea trials data from a data file and formats the data into a 23–term state vector. For vessel data without angle rates, the rates are generated with a MatLab script using simple first–order differentiation:

$$\dot{x}[k] = \frac{x[k] - x[k-1]}{\Delta}. \tag{4.2}$$

The Naval Surface Warfare Center, Carderock Division, supplied maneuvering data recorded during the USCGC HEALY (WAGB 20) Performance and Special Trials from 25-26 August 1999 [Ref. 5]. Figures 4.2 and 4.3 depict the 420-foot twin screw, twin rudder icebreaker. Table 4.2 displays principal vessel characteristics.

The USCGC HEALY (WAGB 20) trials data proved to be an excellent resource. The instrumented vessel has additional sensors not always found on similar sized ships. For example, through the water and over the bottom sway velocity are both available from a Doppler transducer. Appendix E summarizes the parameters available in the data files. Mr. George Brodie of the Naval Surface Warfare Center, Carderock Division, supplied both the data files and a data mining LabView VI. The VI extracts desired parameters and writes them to a tab delimited text file. The Sea Trials Mode of the Data Acquisition module then reads the file and assembles the 23–term state vector. Data are also converted to appropriate units. In general, the states are kept in SI units.

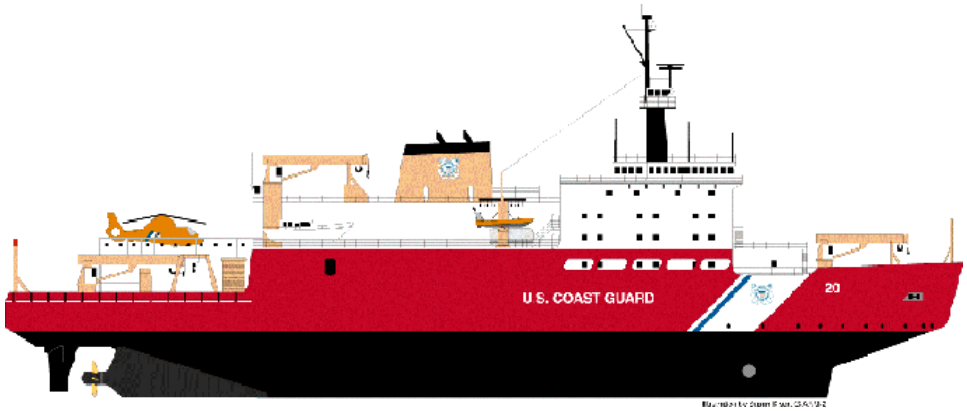


Figure 4.2 Drawing of USCG 420' Icebreaker Class - USCGC HEALY (WAGB 20) [From Ref. 6]

Figure 4.3 Photo of USCG 420' Icebreaker Class - USCGC HEALY (WAGB 20) [From Ref. 7]

Length, Overall	420'0" (128 meters)
Beam, Maximum	82'0" (25 meters)
Draft, Full Load	29'3" (8.9 meters)
Displacement, Full Load	6,000 LT
Propulsion	Diesel Electric
Generating Plant	4 Sultzer 12Z AU40S
Drive Motors	2 AC Synchronous, 11.2 MW
Shaft Horsepower	30,000 Max HP
Propellers	2 Fixed Pitch, 4 Bladed
Auxiliary Generator	EMD 16-645F7B, 2400kW
Fuel Capacity	1,220,915 GAL (4,621,000 liters)
Cruising Speed	12 knots @ 105 RPM
Max Speed	17 knots @ 147 RPM
Icebreaking Capability	4.5 ft @ 3 knots (continuous) 8 ft (2.44 m) Backing and Ramming
Science Labs	Main, Bio-Chemical, Electronics, Meteorological, Photography
Accommodations	19 Officer, 12 CPO, 54 Enlisted, 35 Scientists, 16 Surge, 2 Visitors

Table 4.2 USCGC HEALY (WAGB 20) Principal Characteristics [From Ref. 7]

B. PARAMETER IDENTIFICATION MODULE

1. Parameter Estimation Model Structures

The Parameter Identification Module implements the RLS and ELS algorithms for several model cases. The user selects the cases from the top–level control panel. Table 4.3 summarizes the sway–roll–yaw model structures currently programmed into the module. The same MATLAB parameter estimation script is used for all cases. It implements ELS for the linear coupled sway–roll–yaw model with five terms of noise estimation. Appropriate terms in the regression vector (φ) are zeroed to implement a lesser case. The bias terms (e_{0v}, e_{0p} and e_{0r}) may be individually zeroed from the top control panel. Table 4.3 displays the parameterization and regression vectors for the ELS case. RLS is obtained from the same equations by zeroing the error regression terms and is enabled by a toggle switch on the control panel. All lateral–directional model parameters are non-dimensional by the Prime I ($'$) system.

Table 4.4 summarizes the two RLS surge models. Case #1 assumes a constant surge velocity. This is useful for investigating the various sway–roll–yaw models without cross coupling with the surge equation. Recall that the primary purpose of the parameter identification module is to estimate model coefficients and pass them to the Track Prediction Module. Setting the surge equation to a constant removes one degree of freedom for research purposes and does not pose a large limitation because most vessel maneuvering is performed at a relatively steady forward speed. Backing, collision avoidance and pier–side maneuvering are exceptions. Case #2 implements the non-linear, linear in parameter surge equation. All non-linear surge model parameters are dimensional.

CASE	SWAY-ROLL-YAW MODEL STRUCTURE
1 - ELS	Linear Yaw - Nomoto's 1st Order: $r[k] = f_1\ r[k-1] + g_1\ \delta_R[k-1] + e_{0_r} + e[k] + c_1\ e[k-1] + c_r\ e[k-r]$ $\varphi[k] = [r[k-1]\ \delta_R[k-1]\ 1\ \ \varepsilon[k-1] \ldots\ \varepsilon[k-r]]^T$ $\hat{\theta}[k] = \left[\hat{f}_1 \quad \hat{g}_1 \quad e_{0_r} \quad \hat{c}_1 \ldots \hat{c}_r\ \right]^T$
2 - ELS	Linear Sway–Yaw: $\begin{bmatrix} v[k+1] \\ r[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} v[k] \\ r[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \delta_R[k] + \begin{bmatrix} e_{0_v} \\ e_{0_r} \end{bmatrix} \ldots$ $+ \begin{bmatrix} c_{11}e_v[k-1] & \ldots & c_{1r}e_v[k-r] \\ c_{21}e_r[k-1] & \ldots & c_{22}e_r[k-r] \end{bmatrix}$ $\varphi_1[k] = [v[k-1]\ r[k-1]\ \delta_R[k-1]\ 1\ \ \varepsilon_v[k-1] \ldots \varepsilon_v[k-r]]^T$ $\hat{\theta}_1[k] = \left[\hat{f}_{11} \quad \hat{f}_{12} \quad \hat{g}_1 \quad e_{0_v} \quad c_{11} \quad \ldots \quad c_{1r}\right]^T$ $\varphi_2[k] = [v[k-1]\ r[k-1]\ \delta_R[k-1]\ 1\ \ \varepsilon_r[k-1] \ldots \varepsilon_r[k-r]]^T$ $\hat{\theta}_2[k] = \left[\hat{f}_{21} \quad \hat{f}_{22} \quad \hat{g}_2 \quad e_{0_r} \quad c_{21} \quad \ldots \quad c_{2r}\right]^T$
3 - ELS	Linear Sway–Roll–Yaw: $\begin{bmatrix} v[k+1] \\ p[k+1] \\ r[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \end{bmatrix} \begin{bmatrix} v[k] \\ p[k] \\ r[k] \\ \phi[k] \end{bmatrix} \ldots$ $+ \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \delta_R[k] + \begin{bmatrix} e_{0_v} \\ e_{0_p} \\ e_{0_r} \end{bmatrix} + \begin{bmatrix} c_{11}e_v[k-1] \ldots c_{1r}e_v[k-r] \\ c_{21}e_p[k-1] \ldots c_{2r}e_p[k-r] \\ c_{31}e_r[k-1] \ldots c_{3r}e_r[k-r] \end{bmatrix}$ $\varphi_1[k] = [v[k-1]\ p[k-1]\ r[k-1]\ \phi[k-1]\ \delta_R[k-1]\ 1\ \ \varepsilon_v[k-1] \ldots \varepsilon_v[k-r]]^T$ $\hat{\theta}_1[k] = \left[\hat{f}_{11}\ \hat{f}_{12}\ \hat{f}_{13}\ \hat{f}_{14}\ \hat{g}_1\ \hat{e}_{0_v}\ \hat{c}_{11} \ldots \hat{c}_{1r}\right]^T$ $\varphi_2[k] = [v[k-1]\ p[k-1]\ r[k-1]\ \phi[k-1]\ \delta_R[k-1]\ 1\ \ \varepsilon_p[k-1] \ldots \varepsilon_p[k-r]]^T$ $\hat{\theta}_2[k] = \left[\hat{f}_{21}\ \hat{f}_{22}\ \hat{f}_{23}\ \hat{f}_{24}\ \hat{g}_2\ \hat{e}_{0_p}\ \hat{c}_{21} \ldots \hat{c}_{2r}\right]^T$ $\varphi_3[k] = [v[k-1]\ p[k-1]\ r[k-1]\ \phi[k-1]\ \delta_R[k-1]\ 1\ \ \varepsilon_r[k-1] \ldots \varepsilon_r[k-r]]^T$ $\hat{\theta}_3[k] = \left[\hat{f}_{31}\ \hat{f}_{32}\ \hat{f}_{33}\ \hat{f}_{34}\ \hat{g}_3\ \hat{e}_{0_r}\ \hat{c}_{31} \ldots \hat{c}_{3r}\right]^T$

Table 4.3 Sway–Roll–Yaw Parameter Identification Model Structures

CASE	SURGE EQUATION MODEL STRUCTURE
1 - RLS	Constant Surge: $u[x] = u[x-1]$
2 - RLS	Non-Linear Surge - Linear in Parameters: $u[k+1] = f_{u1}\, u[k] + f_{u2}\, \|u\|u[k] + f_{u3}\, v[k]r[k] + f_{u4}\, r[k]^2 \ \ldots$ $+ g_{u1}\, \delta_R[k]^2 + g_{u2}\, \|n\|n[k] + e_{0u}$ $\varphi_u[k] = \left[u[k-1] \ \ \|u\|u[k-1] \ \ v[k-1]r[k-1] \ \ r[k-1]^2 \ \ \delta_R[k-1]^2 \ \ \|n\|n[k-1] \ \ 1\right]^T$ $\hat{\theta}_u[k] = \left[\hat{f}_{u1} \ \hat{f}_{u2} \ \hat{f}_{u3} \ \hat{f}_{u4} \ \hat{g}_{u1} \ \hat{g}_{u2} \ \hat{e}_{0u}\right]^T$

Table 4.4 Surge Parameter Identification Model Structures

2. Eigenvalue Calculations

In addition to estimating model parameters, the Parameter Identification Module calculates and plots sway–roll–yaw plant eigenvalues in real–time. Observing eigenvalue convergence and stability provides a better indication of parameter identification progress than observing a long list of individual parameters.

It is a trivial task to calculate eigenvalues for continuous–time models using any of several software packages, including LabView or MatLab. However, the parameter identification model estimates the coefficients for a discrete–time model. The discrete–time plant model is converted back to a continuous–time plant to perform the eigenvalue calculation. The linear, sway–roll–yaw model is the most complex of the models tested, so it will be used to demonstrate the process.

Recall the continuous–time sway–roll–yaw model with heading integration:

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix} \delta_R \ . \tag{4.3}$$

The discrete–time state–space representation of the system is:

$$x[k+1] = F\, x[k] + G\, u[k] \tag{4.4}$$

where:

$$F = e^{A\Delta} \text{ and } G = A^{-1}(F - I)B. \tag{4.5}$$

Provided Δ is small, the ZOH (Euler discretization) system approximation may be used:

$$F \approx I + A\Delta \quad \text{and} \quad G \approx B\,\Delta, \tag{4.6}$$

yielding a discrete plant matrix F:

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ 0 & \Delta & 0 & 1 \end{bmatrix}. \tag{4.7}$$

Roll integration is known a priori, so there is no need to verify it through parameter identification. The reduced parameter identification model becomes 3 x 4 with the roll integration removed:

$$\begin{bmatrix} v[k+1] \\ p[k+1] \\ r[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \end{bmatrix} \begin{bmatrix} v[k] \\ p[k] \\ r[k] \\ \phi[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \delta_R[k] + \begin{bmatrix} e_v \\ e_p \\ e_r \end{bmatrix}. \tag{4.8}$$

After the parameters are identified, the model is augmented back to the square (4 x 4) discrete plant matrix, *F*, so the continuous–time plant matrix, *A*, may be calculated:

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ 0 & \Delta & 0 & 1 \end{bmatrix}. \tag{4.9}$$

The parameter estimation is performed with non-dimensional states to reduce parameter sensitivity to vessel speed. In other words, the parameter estimation module is actually estimating F' according to the Prime – I system. So, the continuous–time plant matrix, *A*, is time scaled to obtain dimensional eigenvalues:

$$A = \frac{\text{logm}\left(F'\right)}{\Delta} \cdot \frac{L}{U}. \tag{4.10}$$

3. Non-Dimensional Time Scaling

Vessel parameter estimation is best performed with non-dimensional states to minimize vessel velocity effects. As stated earlier, the parameter estimation module is actually estimating F' according to the Prime – I system. An unintended outcome is a time scaling problem. As written, the RLS and ELS equations assume a constant sampling interval. However, in this case the fixed, dimensional sampling interval is speed–dependent in the Prime – I as follows:

$$\Delta' = \Delta \frac{U}{L}. \tag{4.11}$$

The basic RLS/ELS equations work by generating an error between an estimated non-dimensional state and the actual non-dimensional state using the ZOH approximation:

$$\varepsilon'(k) = y'(k) - \varphi'^{T}(k)\,\hat{\theta}'(k-1). \tag{4.12}$$

If the RLS/ELS equations are not adjusted for sample interval speed dependence, the estimated non-dimensional parameters are speed–dependent. The author chose to work around the problem by fixing the non-dimensional sampling interval and then time scaling the parameter estimates to correspond to the actual sampling interval. The parameters are scaled prior to calculating the estimation error (Equation 4.12) using the following technique:

$$\hat{\theta}'_{\,time\ scaled} = \left(\hat{\theta}' - [1, 0, 0 \ldots, 0_{n+m}]\right)\frac{U_0}{U} + [1, 0, 0 \ldots, 0_{n+m}] \tag{4.13}$$

where U_0 is the average service speed and U is the actual speed.

Another way to get around the time scaling problem is to implement a log scaled, speed–dependent sampling interval directly into the RLS/ELS equations. This would provide more robust time scaling and reduce speed dependency caused by the ZOH approximation. Unfortunately, the author discovered the time scale problem late in devel-

opment and used the work around listed above rather than completely re-write the ELS code.

4. Current Estimation

The Parameter Estimation Module provides estimated current to the Path Prediction Module. Current is obtained by subtracting measured "through the water" velocity from the inertial velocity. This is most easily accomplished by transforming the water velocity to the NED tangent plane:

$$V_{Ci} = V_i - R_{b-i}V_{Wb} \tag{4.14}$$

where V_{Ci} is the current in the NED inertial reference plane, V_i is the vessel inertial velocity in the NED reference plane, $R_{b\text{-}i}$ is the body axis to NED rotation matrix and V_{Wb} is the vessel water velocity measured in the body axis system

The current estimate is filtered by a simple five–parameter moving–average filter. In addition, current estimation is limited to periods of low turn rate and low rudder deflection to reduce errors caused by inaccurate sideslip measurement.

C. PATH PREDICTION MODULE

The Path Prediction Module performs current adjusted, open–loop integration to estimate future path over the ground. An array of track points is passed to the Track Display Module. Final vessel heading is also passed so that a scaled and rotated vessel outline may also be displayed at the path prediction endpoint.

When ELS is selected, the initial integration time step uses ELS filtering to provide the best estimate of the initial states. For instance, integration for the yaw rate model is initiated with:

$$y(k) = \varphi^T(k)\,\hat{\theta}(k-1) \tag{4.15}$$

where:

$$\varphi[k] = \left[r[k-1]\ \delta_R[k-1]\ 1\ \varepsilon[k-1] \ldots\ \varepsilon[k-r] \right]^T \tag{4.16}$$

$$\hat{\theta}[k] = \left[\hat{f}_1 \quad \hat{g}_1 \quad e_{0r} \quad \hat{c}_1 \ldots \hat{c}_r \right]^T. \tag{4.17}$$

Subsequent time steps are integrated using the model parameters and regression vector without the filter terms ($C\left(z^{-1}\right)$):

$$\varphi[k]=\left[\, r[k-1]\;\delta_R[k-1]\;\;1\,\right]^T \tag{4.18}$$

$$\hat{\theta}[k]=\left[\,\hat{f}_1 \quad \hat{g}_1 \quad e_{0r}\,\right]^T. \tag{4.19}$$

Note that bias terms are still included because they continue to compensate for sensor biases and environmental disturbances throughout the integration interval.

The Path Prediction Module also captures the maximum left and right roll angles encountered during integration and passes them to the Track Display Module for display. This is a useful function for carriers maneuvering during aircraft handling or flight operations.

D. TRACK DISPLAY MODULE

The Track Display Module presents the path estimate to the vessel bridge team in a graphical format. Figure 4.4 shows a snapshot of the primary system display. Indicators are provided for rudders, shaft speed, heading, etc. Note that many additional controls are also placed on this top–level display. This is purely for developmental proposes and only a few selections would be available to the user in an actual implementation.

An estimated path-over-the-ground line is overlaid in real–time on the top–level ECN image. In addition, a scaled and rotated vessel outline is displayed at the final estimated position. The primary project emphasis was on parameter identification and not the development of an interactive electronic chart viewer. So, the simulation is somewhat crude but sufficient to demonstrate the overall concept.

ECN images were obtained from NOAA's ECN databases. The databases are available for free download at http://chartmaker.ncd.noaa.gov/mcd/enc/download.htm. An ECN viewer is required to process the database files into an electronic chart. A free viewer called "SeeMyDEnc" was downloaded from the Seven C's website at http://www.sevencs.com. After centering and zooming, an ECN image is exported as an

Windows bitmap (".bmp") file. The latitude and longitude of corner points is also required to appropriately scale the image.

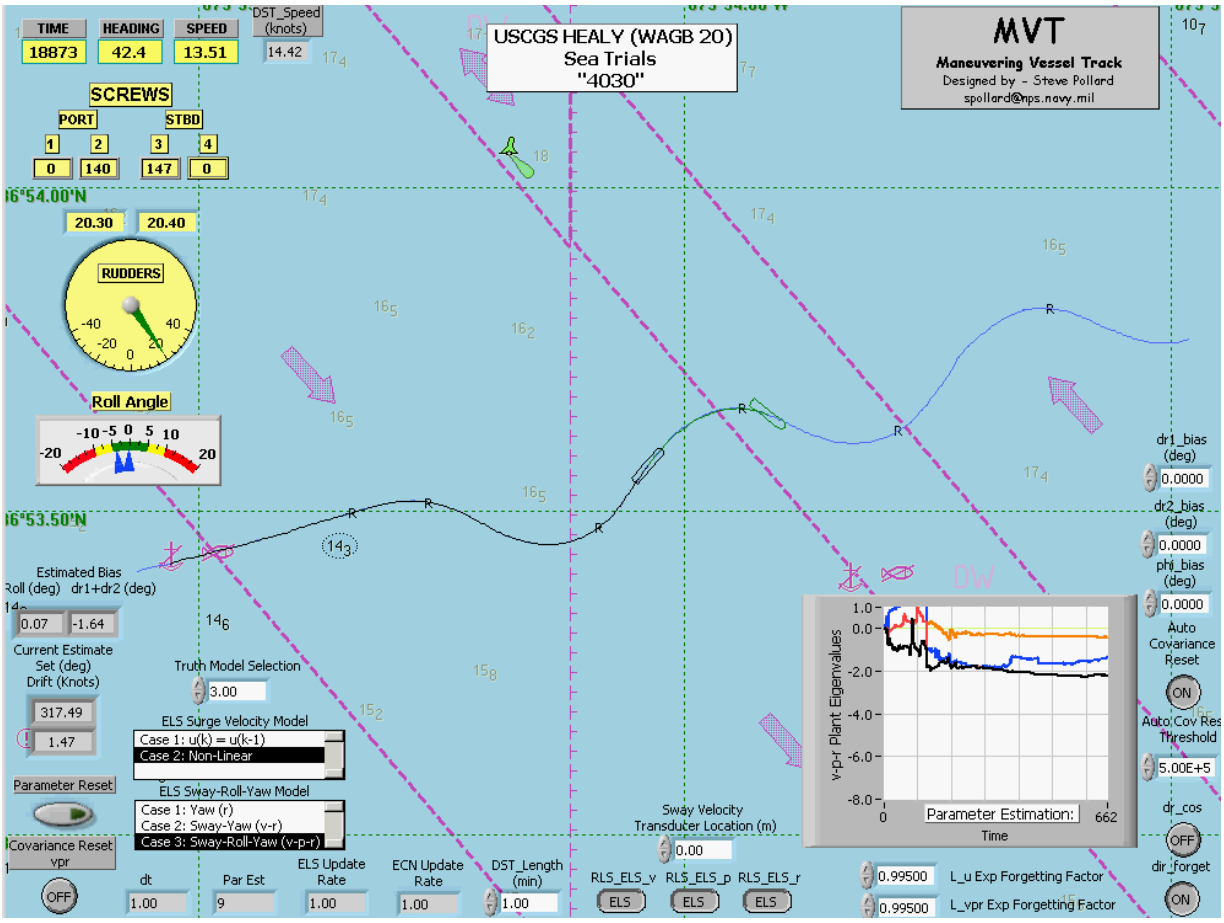


Figure 4.4 Top Level Controls and Display

One of the reasons LabView was selected as the overall software platform is a rich graphical user interface. LabView contains built-in capability to read the popular PC image files and then draw graphics on top of the image by manipulate pixels. In this case, the primary functionality used was the "draw multiple lines" on a picture VI. The path estimation and vessel outline data points are scaled and mapped to individual pixels. The "draw multiple lines" VI is then called to draw the lines and display the image.

In addition to estimated track data, a breadcrumb trail, or past track, is left behind the vessel. When a recorded file is the data source, the system can overlay the track for

the entire file at start-up. This is particularly useful for comparing the estimated path to actual path.

E. CHAPTER SUMMARY

This chapter stepped the reader through the overall path prediction system design. The Data Acquisition Module obtains and formats vessel state vector data from selectable math models or from previously recorded sea trials data. The vessel data is passed to the Parameter Identification Module where the model parameters are recursively estimated. The model parameters are used by the Path Prediction Module to estimate future vessel path over the ground. Vessel path is estimated through open–loop integration of the vessel model using the current state observation as the initial condition. Finally, the predicted path is passed to the Track Display Module for overlay on an electronic chart image. Results are presented in Chapter V in a build-up approach. System functionality is validated prior to presenting full system performance with real–world sea trials data.

V. RESULTS

Results are presented in a build up approach with the final goal being to demonstrate overall system performance while processing real–world sea trials data. Parameter estimation functionality is tested in a noise free environment prior to proceeding to the noisy sea trials data. The general flow of the results section is as follows:

- Verify the steering (*v–p–r*) Parameter Estimation Module RLS/ELS algorithms using math model data.
- Investigate covariance wind–up and numerical stability.
- Compare the relative merits of the linear sway–roll–yaw models.
- Compare linear model matching to non-linear truth data.
- Verify current and bias algorithms.
- Demonstrate linear and non-linear surge model performance.
- Demonstrate end-to-end path prediction in a noisy environment using sea trials data.

A. STEERING PARAMETER ESTIMATION MODULE VERIFICATION

The lateral–directional (*v–p–r*) parameter estimation implementation was confirmed using linearized containership truth model data. All three RLS/ELS vessel models demonstrate stable convergence in a noise free environment.

A standard series of rudder input excitation was applied to each model for approximately 300 seconds. Identical control inputs allow direct comparison of results. The system was initialized to the settings in Table 5.1. Exponential forgetting was required to discount start-up transients and obtain parameter convergence in a reasonable time.

Parameter	*Setting*
Truth Data	Linearized Containership Model – Uo = 7m/s
Truth Model Time Step	0.2 s (RK2 Integration)
RLS/ELS – *v-p-r* Model	ELS with *v*, *p* and *r* Bias Enabled
RLS/ELS Time Step	0.2 s
Exponential Forgetting	0.99 for *v-p-r* Model
Directional Forgetting	On
Covariance Reset	Auto @ cond $P_{vpr} > 0.5 \times 10^6$, $P_{vpr} = P_{vpr} + 10*I$
Surge Model	Constant Surge Speed ($u[k] = u[k-1]$)

Table 5.1 Settings for Parameter Estimation Module Verification

1. Linearized Yaw Rate Model

The yaw rate only model was verified with yaw rate truth data from the linearized Fossen containership model (Table 4.1 – Case 1). Figure 5.1 shows the rudder excitation sequence and resulting vessel states. Note that only yaw rate is excited with this simplified model. Figure 5.2 demonstrates plant and control parameter convergence. The RLS/ELS algorithm is estimating one plant and one control parameter for a total of two parameters. Figure 5.3 illustrates yaw mode eigenvalue convergence. Note that the parameters and eigenvalues tend to be slightly faster than in the truth model. This is due to the ZOH approximation used in the discrete to continuous–time conversion. A longer non-dimensional sample time (Δ') increases this effect.

2. Linearized Sway–Yaw Model

The linearized sway–yaw model was verified with truth data from the linearized Fossen containership model (Table 4.1 – Case 2) in a similar fashion to the yaw rate model. Figure 5.4 shows the rudder sequence used to excite the vessel states. Notice that both sway velocity and yaw rate are now excited. Figure 5.5 demonstrates plant and con-

trol parameter convergence. The RLS/ELS algorithm is now estimating four plant and two control parameters for a total of six parameters. Figure 5.6 illustrates sway and yaw mode eigenvalue convergence. The incorporation of the much slower sway mode causes parameter identification convergence to slow as well. Initial sway–yaw convergence occurs in approximately 65 seconds versus 30 seconds for just the yaw mode. Again, the parameters and eigenvalues tend to be slightly faster than the truth model due to the ZOH approximation used in the discrete to continuous–time conversion.

3. Linearized Sway–Roll–Yaw Model

The linearized sway–roll–yaw model was verified with truth data from the linearized Fossen containership model (Table 4.1 – Case 3). Figure 5.7 shows the rudder sequence and associated vessel states. Notice that all lateral-directional states are excited with the *v–p–r* model. Figure 5.8 demonstrates plant and control parameter convergence. Switching from the sway–yaw (*v–r*) model to the sway–roll–yaw (*v–p–r*) doubles the order to a 4x4 plant model, including roll angle integration. This results in the estimation of twelve plant and three control parameters. Convergence occurs in approximately 250 seconds versus 30 seconds for the 1x1 yaw plant model and 65 seconds for the 2x2 sway-yaw plant model. The convergence rate is roughly proportional to the size of the model squared.

Figure 5.9 illustrates sway, roll and yaw mode eigenvalue convergence. The lightly damped roll mode is demonstrated by two eigenvalues in the polar plot. Notice that during the convergence process, the roll mode makes two trips to the unstable, right hand plane. This is more pronounced when the vessel states are excited by a sea state, as some wave sequences will make the roll mode appear unstable. As with the other models, the parameters and eigenvalues tend to be slightly faster than the truth model due to the ZOH approximation used in the discrete– to continuous–time conversion.

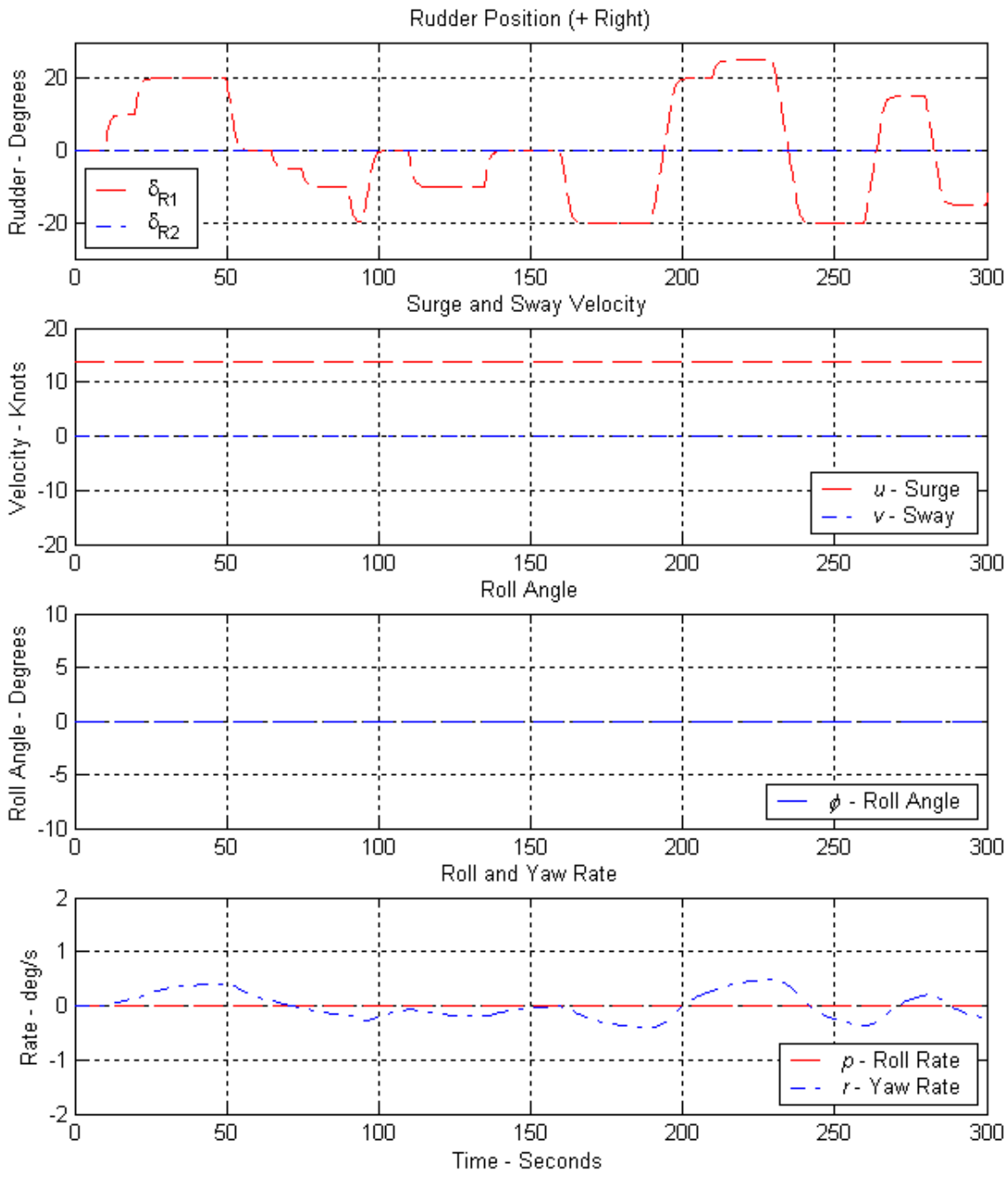


Figure 5.1 Yaw Rate Model Excitation

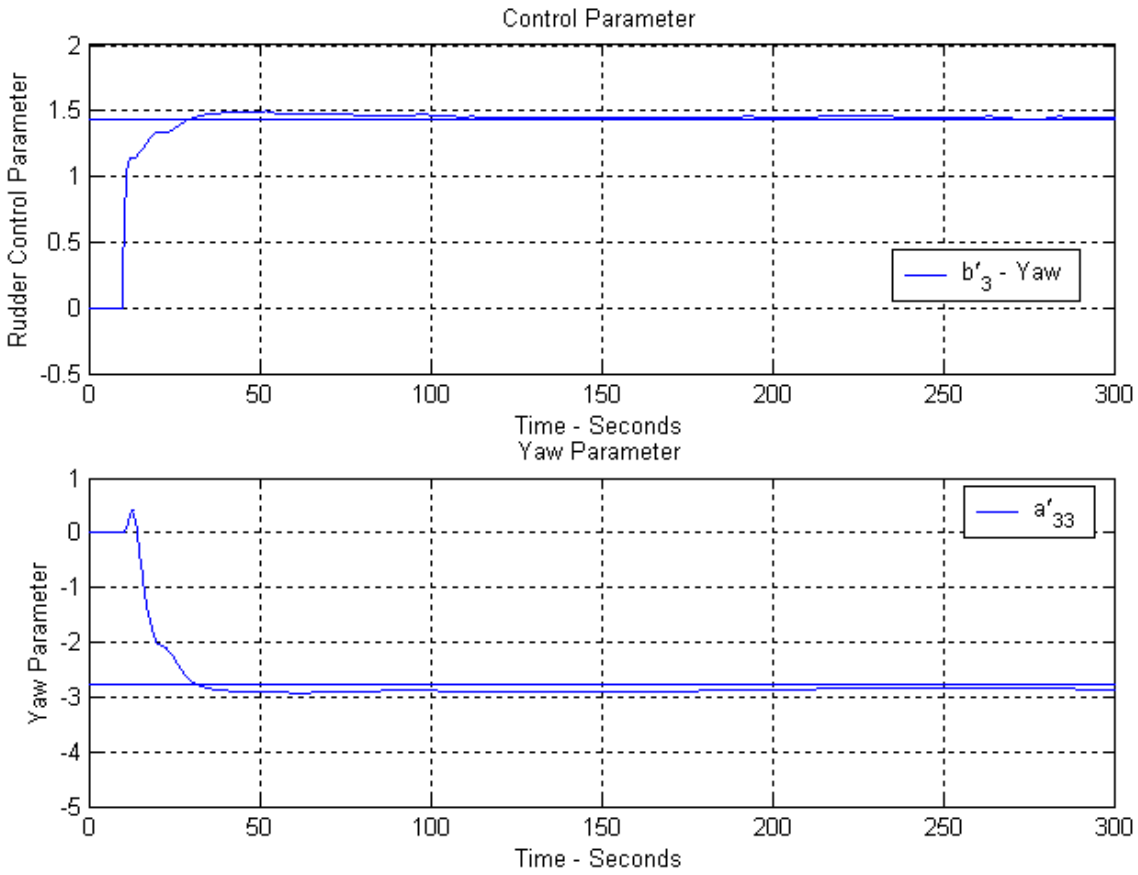


Figure 5.2 Yaw Rate Model Parameter Convergence

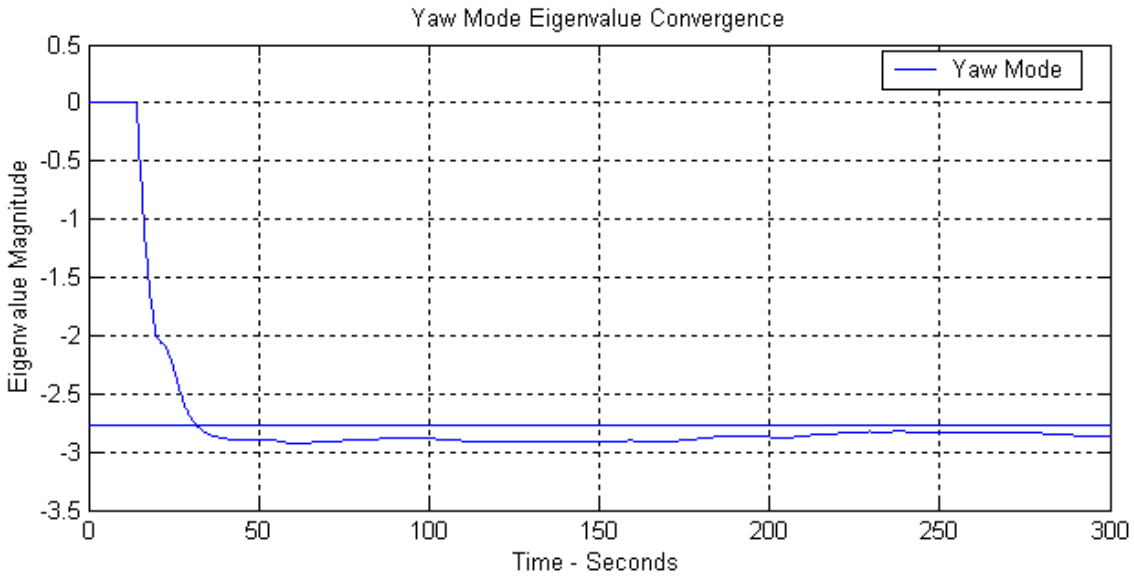


Figure 5.3 Yaw Rate Model Eigenvalue Convergence

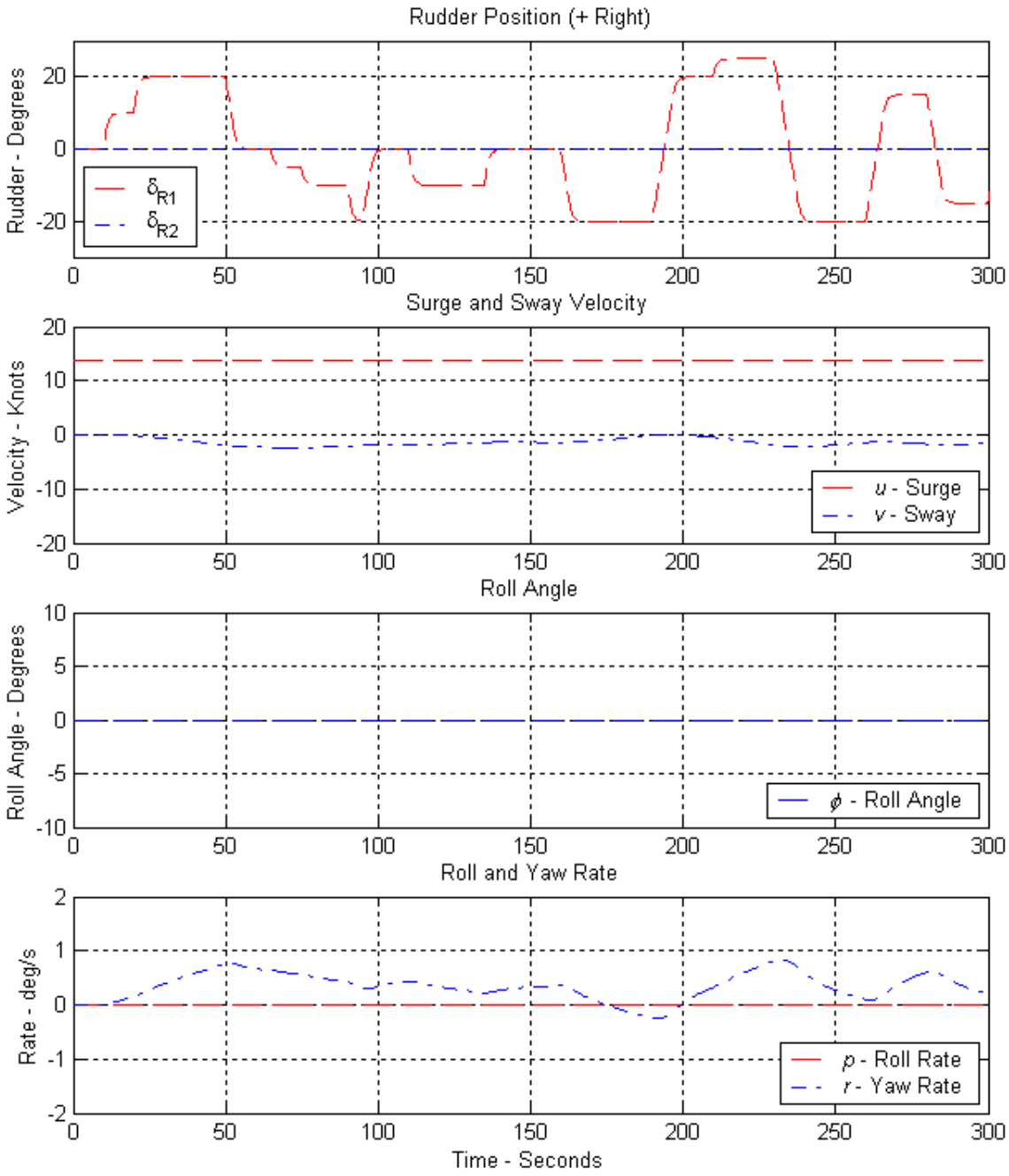


Figure 5.4 Sway–Yaw Model Excitation

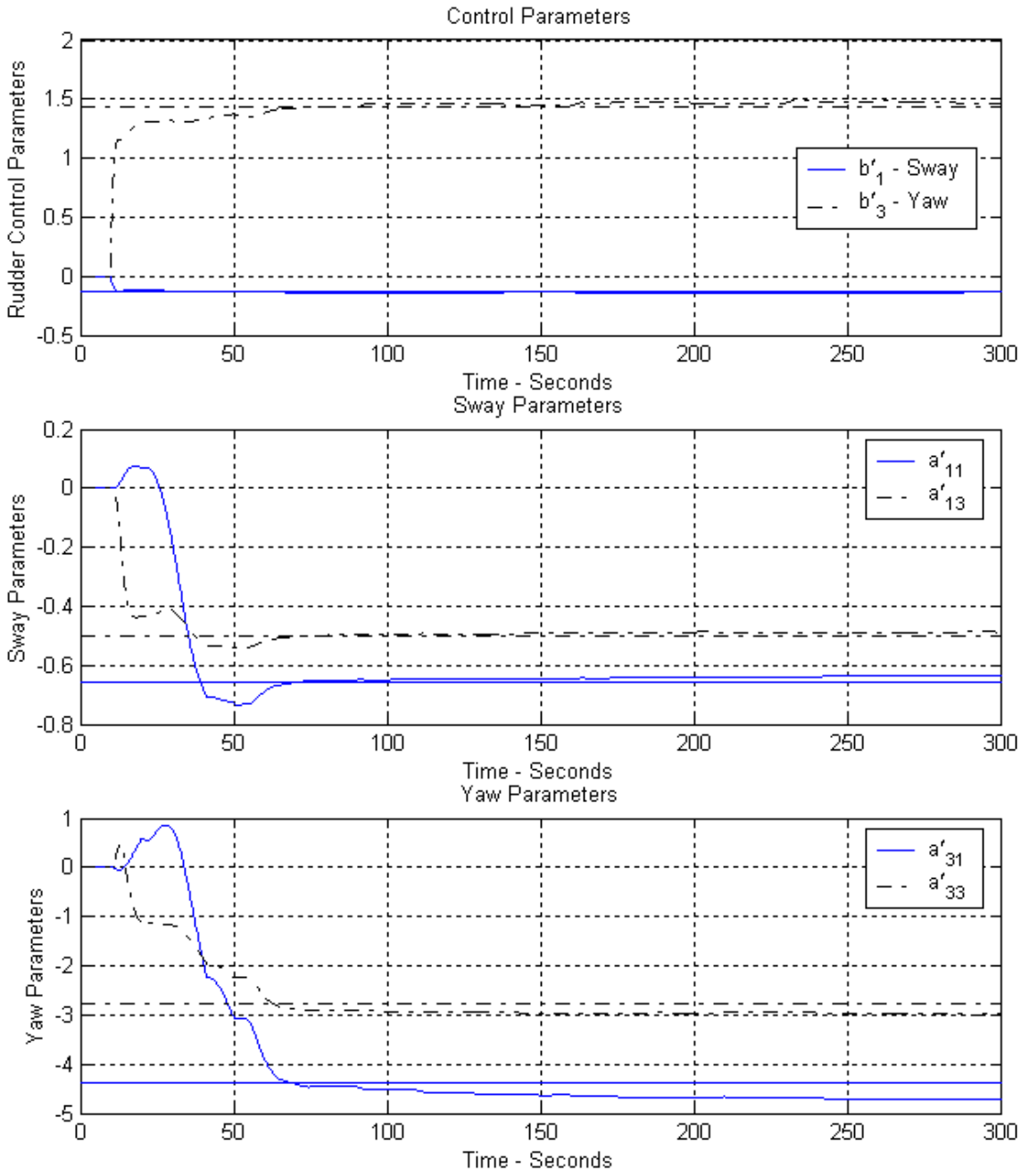


Figure 5.5 Sway–Yaw Model Parameter Convergence

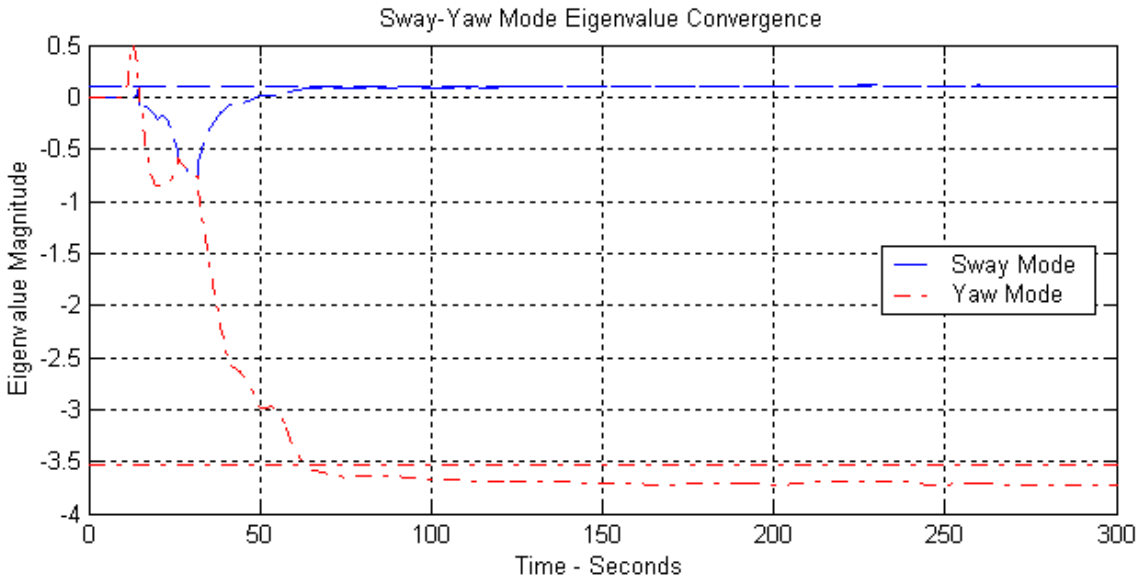


Figure 5.6 Sway–Yaw Model Eigenvalue Convergence

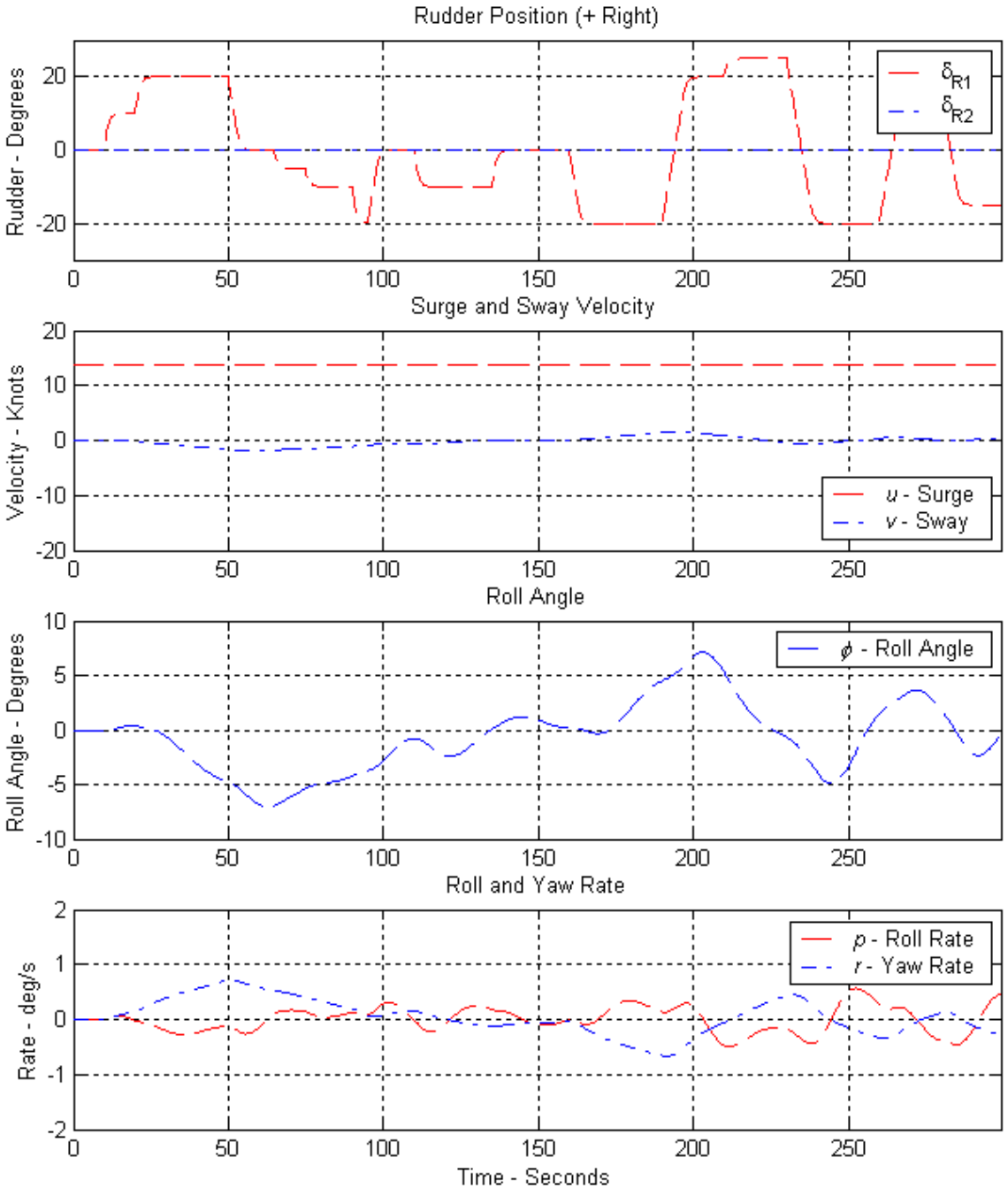


Figure 5.7 Sway–Roll–Yaw Model Excitation

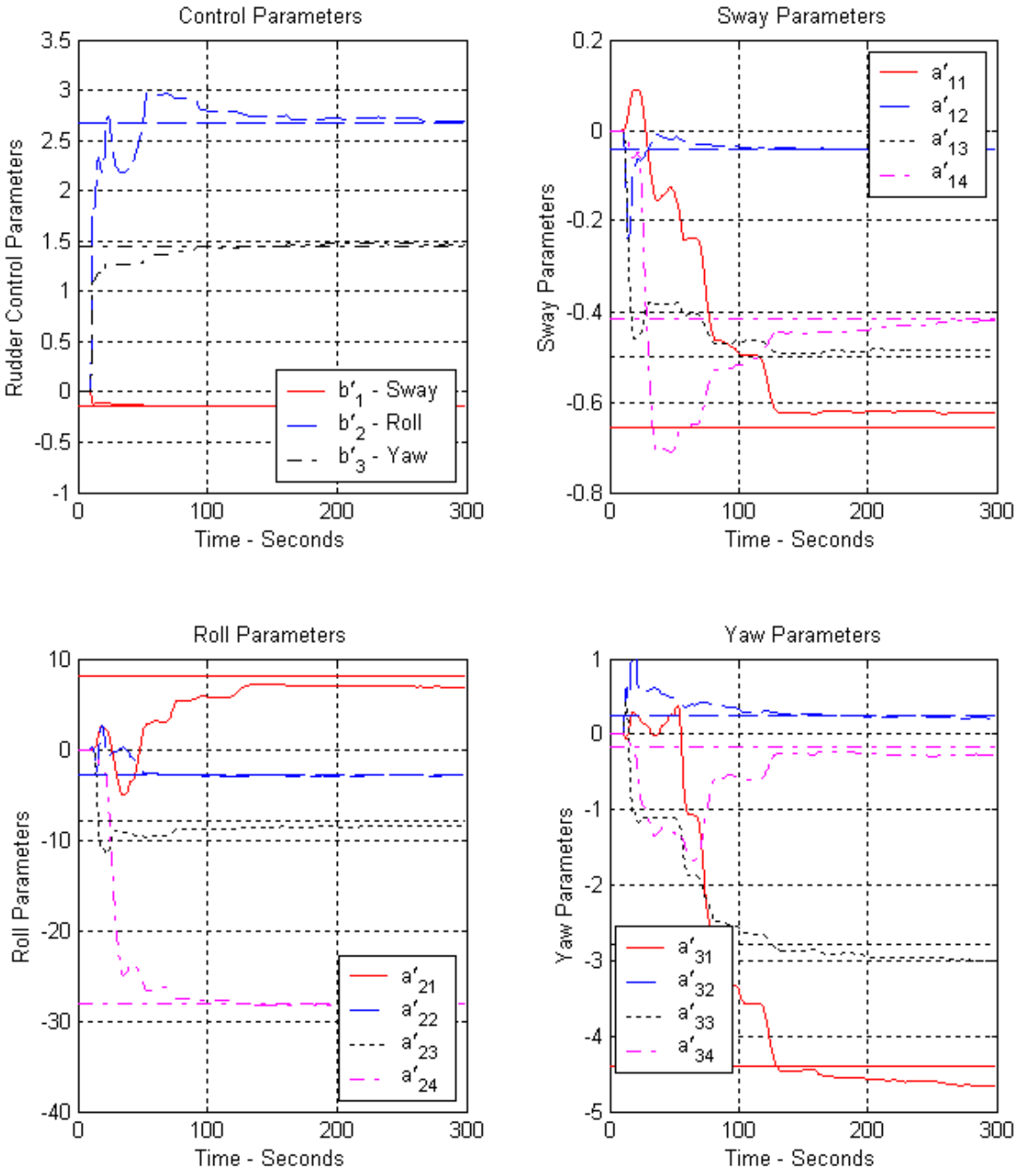


Figure 5.8 Sway–Roll–Yaw Model Parameter Convergence

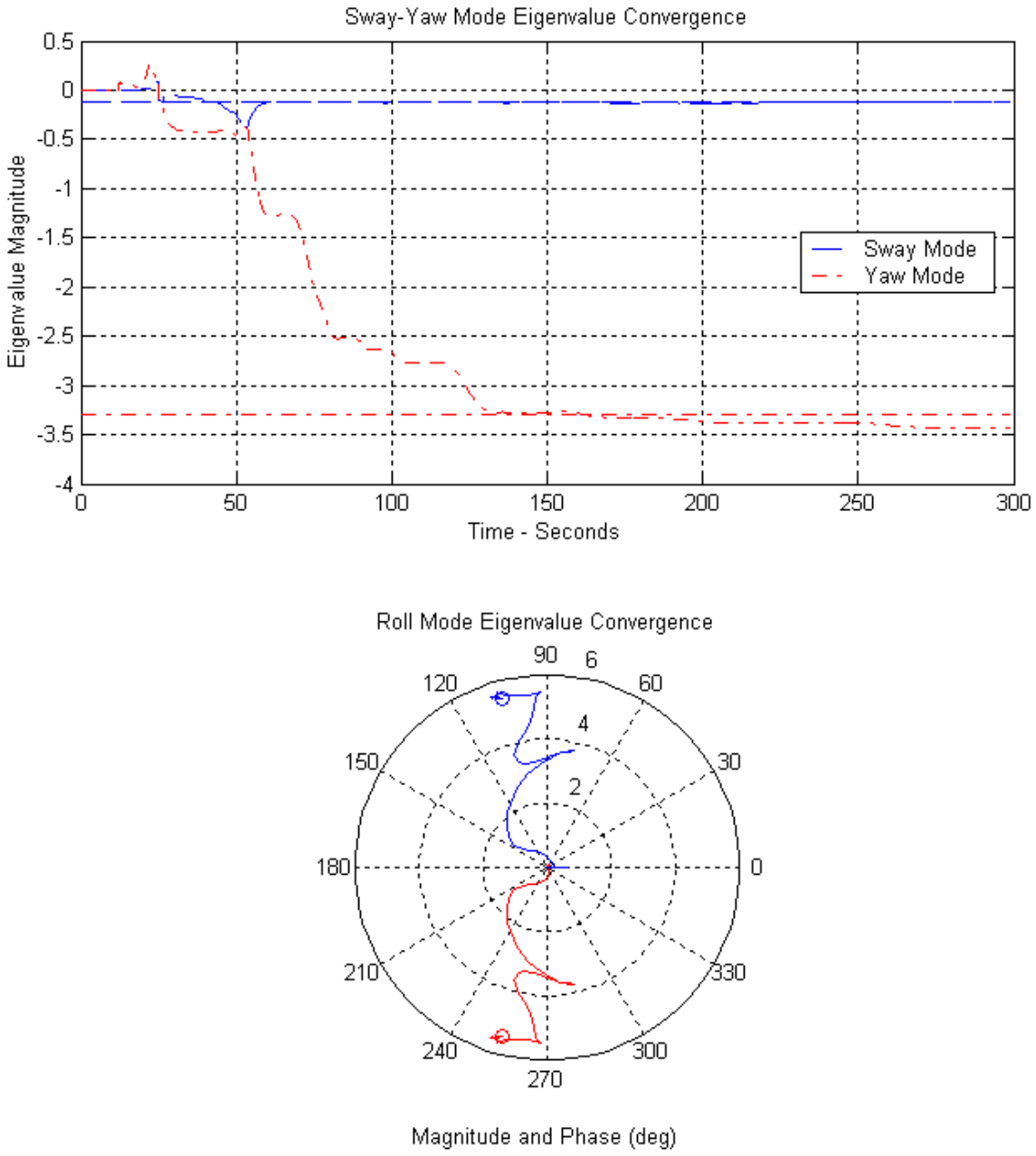


Figure 5.9 Sway–Roll–Yaw Model Eigenvalue Convergence

B. INVESTIGATION OF COVARIANCE WIND–UP

Exponential forgetting was implemented to discount start-up transients and capture time–varying parameters. However, exponential forgetting causes covariance wind–up during periods of insufficient excitation. Even if there is sufficient excitation from some states to keep a portion of the covariance matrix bounded, the under–excited parameters will still cause covariance wind–up. Standard exponential forgetting is applied evenly to all columns of the covariance matrix update, causing exponential growth in the unexcited portion of the covariance matrix.

Two techniques are implemented to deal with covariance wind–up. One is conditional updating and the other is directional forgetting combined with automatic covariance resetting. Implementation of conditional updating is straightforward. The parameter estimation algorithm updates only when sufficient excitation is present to contribute to the parameter estimation solution. The Parameter Estimation Module conditionally updates when:

$$\varphi(t)^T P(t)\varphi(t) > 2(1-\lambda). \tag{5.1}$$

Directional forgetting proved more difficult to implement. The matrix inverse operation in the directional forgetting algorithm (Equation 3.62) creates numerical instability when the covariance matrix is ill conditioned. To prevent instability, the covariance matrix is reset when the condition number exceeds a preset threshold. The most effective reset seems to be the addition of a small identity matrix. The combination of conditional updating and directional forgetting with automatic covariance reset performed well. The next section illustrates the progression of techniques implemented to control covariance wind–up. Table 5.2 lists the settings and Figure 5.10 displays the excited states for all three scenarios.

Parameter	***Setting***
Truth Data	Linearized Sway-Roll-Yaw Containership Model – Uo = 7 m/s
Truth Model Time Step	0.2 s (RK2 Integration)
RLS/ELS – *v-p-r* Model	ELS with *v*, *p* and *r* Bias Enabled
RLS/ELS Time Step	0.2 s
Exponential Forgetting	0.99 for *v-p-r* Model
Covariance Reset	Off or Auto @ cond $P_{vpr} > 0.5 \times 10^6$, $P_{vpr} = P_{vpr} + 10*I$
Surge Model	Constant Surge Speed ($u[k] = u[k-1]$)

Table 5.2 Initialization Settings for Covariance Wind–Up Investigation

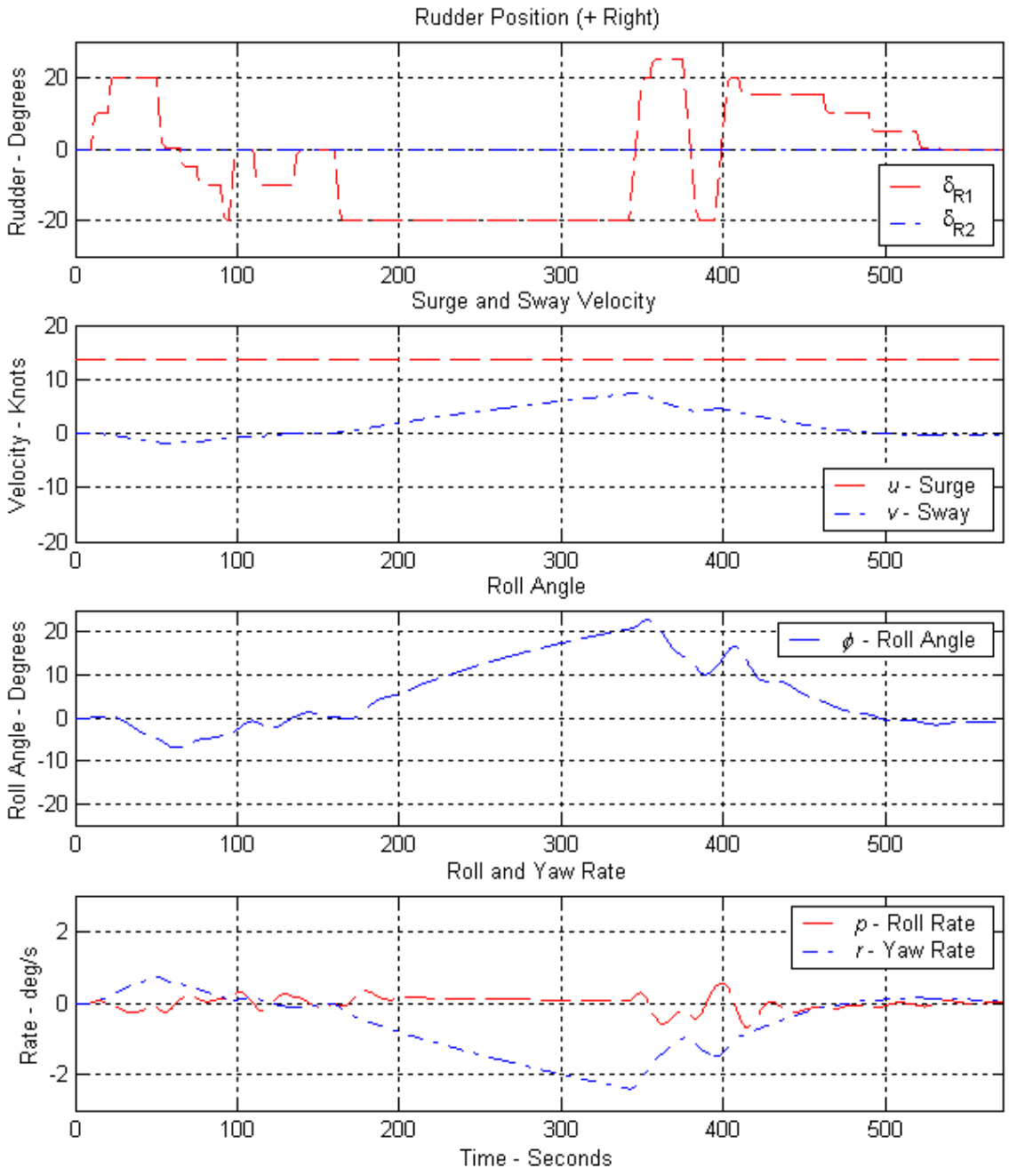


Figure 5.10 Vessel State Excitation During Covariance Wind-Up Investigation

1. Exponential Forgetting Induced Covariance Wind–Up

Figure 5.11 displays the trace and condition number of the sway–roll–yaw covariance matrices with standard exponential forgetting. The covariance matrix displays classic wind–up due to lack of excitation of all the parameters. It is important to note that standard exponential forgetting acts evenly across all parameters. Thus, the covariance

matrix winds up even with just one unexcited parameter. In this case, the noise parameter terms are largely unexcited and display exponential growth. After 300 seconds, the algorithm displays numerical instability. The condition number of the covariance matrix also grows exponentially with the trace.

2. Covariance Wind-Up Mitigated by Directional Forgetting

Figure 5.12 displays the trace and condition number of the sway–roll–yaw covariance matrices with directional forgetting. Covariance growth is well behaved. However, inversion of the covariance matrix causes instability at 103 seconds and again at 220 seconds. Interestingly, the instability recovers after the episode at 103 seconds. The second episode at 220 seconds crashed the calculations.

3. Directional Forgetting Combined with Covariance Resetting

Figure 5.13 displays the trace and condition number of the sway–roll–yaw covariance matrices with directional forgetting and automatic covariance reset enabled. Several runs were conducted with different excitation to determine a suitable covariance reset criteria. Numerical stability seemed assured when the condition of P was kept less than $0.5\text{x}10^6$. A suitable reset value was also needed. The first runs reset the covariance matrix to the initial value of $5000*I$. The resulting high gain caused drift in already converged parameters. Another method is to add a small identity matrix. This approach worked well. In Figure 5.13, the covariance matrix was reset to $P = P + 10 * I$.

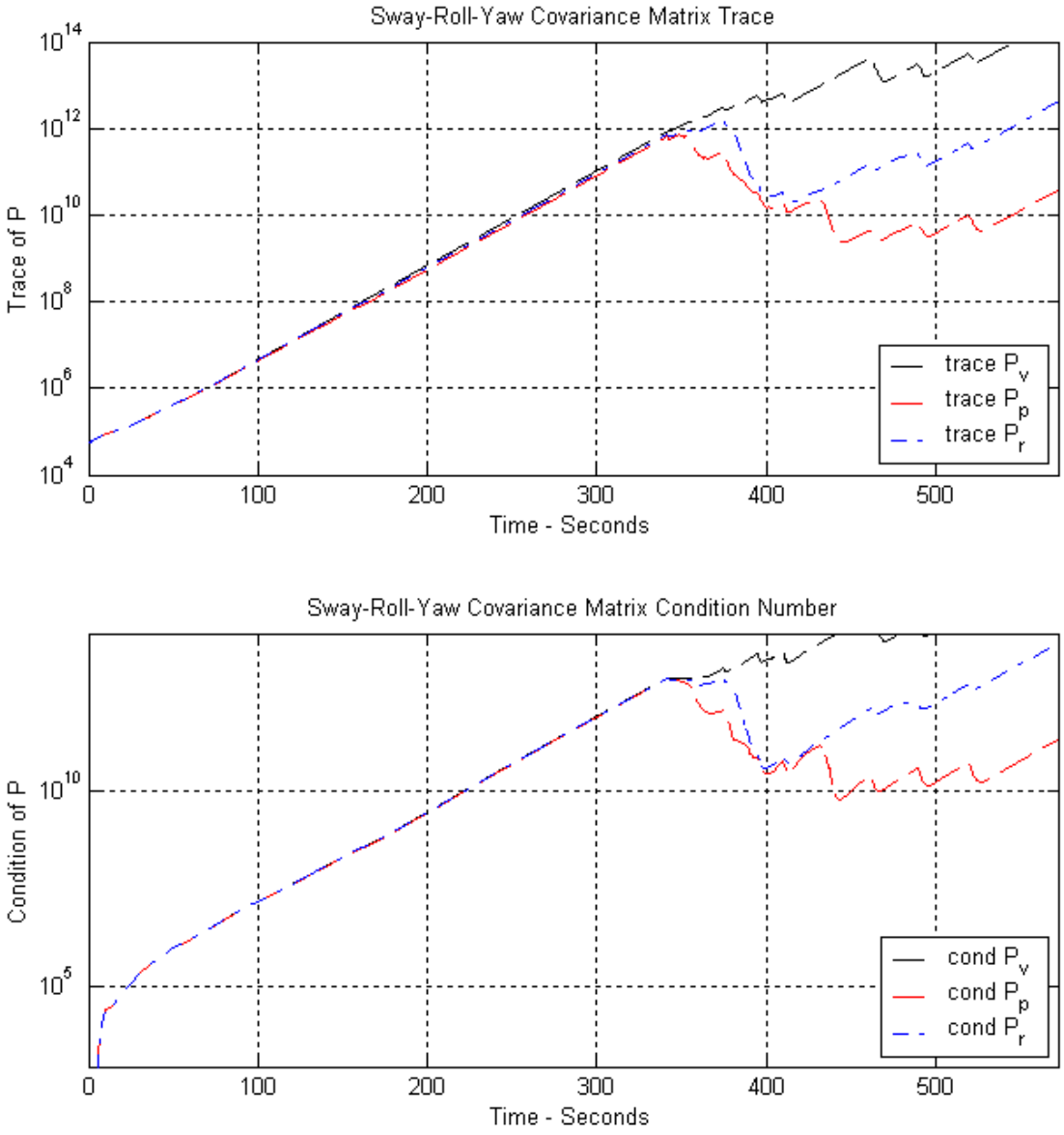


Figure 5.11 Trace and Condition Number of Covariance Matrices During Standard Exponential Forgetting ($\lambda = 0.99$)

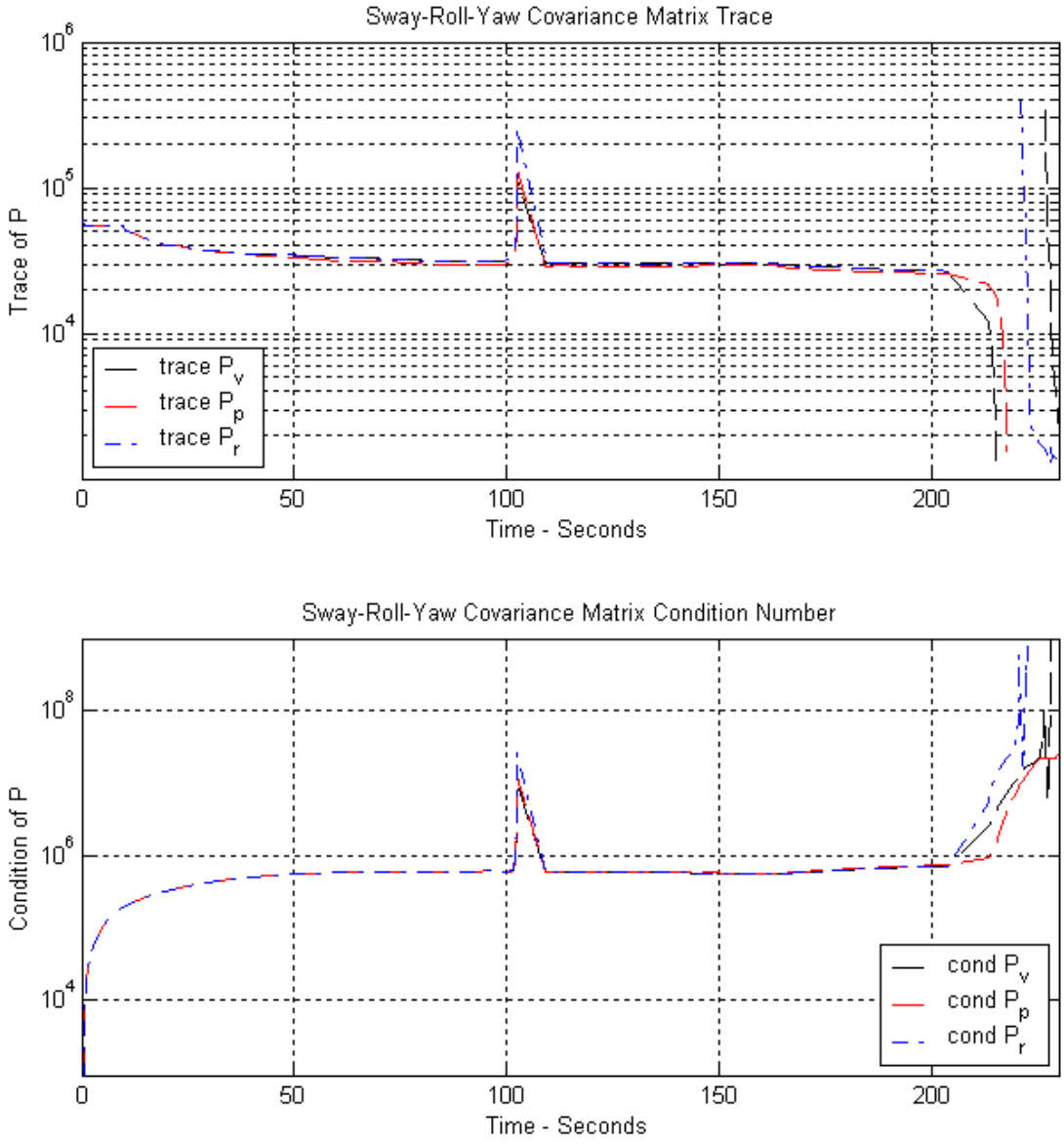


Figure 5.12 Trace and Condition Number of Covariance Matrices During Directional Forgetting without Covariance Reset ($\lambda = 0.99$)

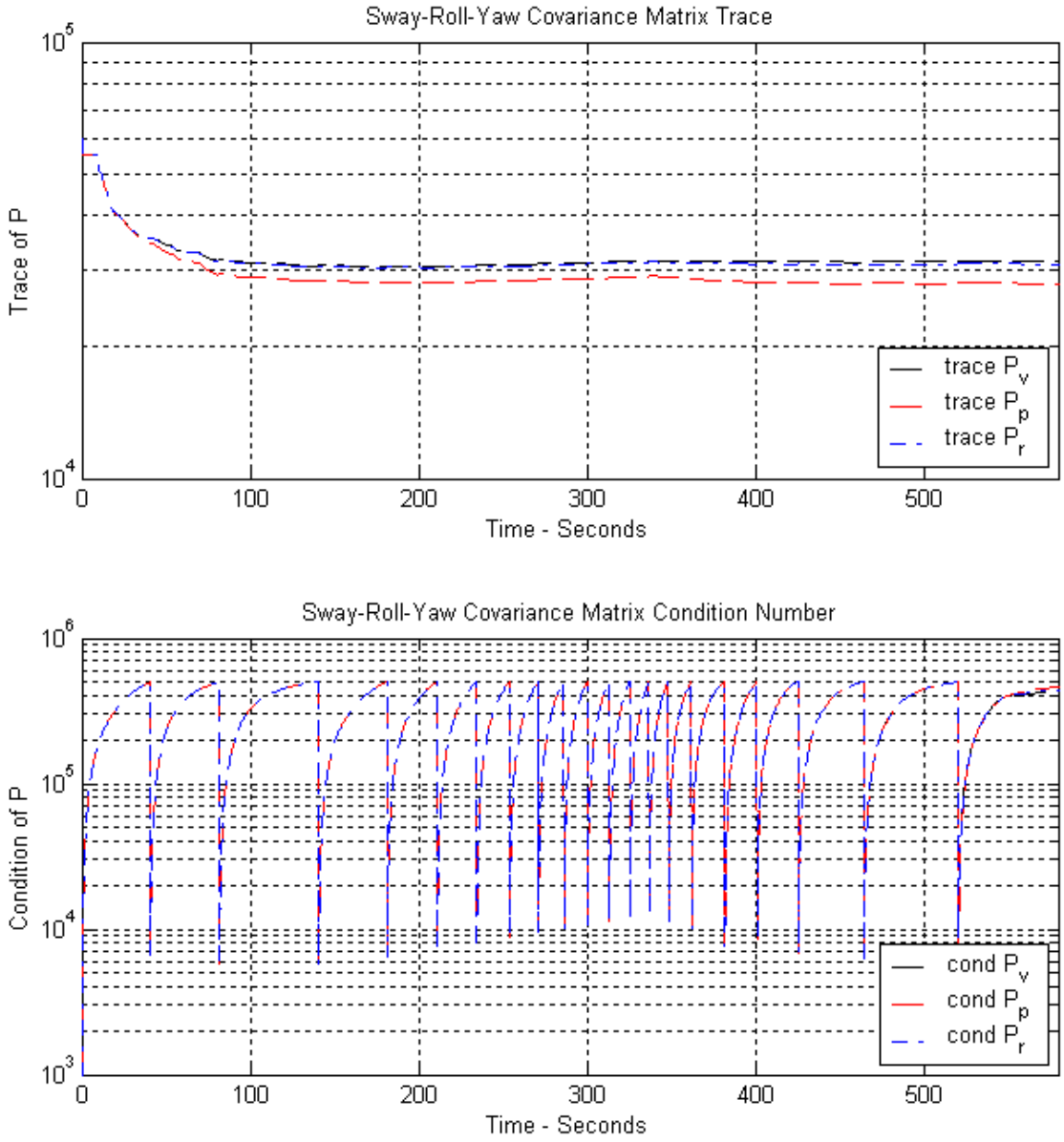


Figure 5.13 Trace and Condition Number of Covariance Matrices During Directional Forgetting with Covariance Reset @ Cond (P) > 0.5x10^6 ($\lambda = 0.99$)

C. INVESTIGATION OF SIMPLIFIED SWAY–ROLL–YAW MODELS

Section A verified the computational integrity of the three linear steering models and Section B investigated methods for mitigating covariance wind–up. Now, suitability of linear steering models for maneuvering path prediction is investigated.

Parameter estimation stability and path prediction snapshots were compared for the yaw rate (r), sway–yaw (v–r) and sway–roll–yaw (v–p–r) models. All three models were excited by identical truth data from the linearized containership model. Both the sway–yaw and sway–roll–yaw models demonstrated sufficient fidelity for maneuvering path estimation. The yaw rate model was found to be only suitable for low bandwidth control inputs that generate a nearly constant turn rate.

A standard series of rudder input excitation was applied to all three models for approximately 500 seconds. Exponential forgetting was enabled. Identical control inputs and state excitation allows direct comparison of results. The system was initialized to the settings in Table 5.3. Figure 5.14 displays the control inputs and states that stimulated the parameter estimation models.

Parameter	***Setting***
Truth Data	Linearized Containership Model – Uo = 7 m/s
Truth Model Time Step	0.2 s (RK2 Integration)
RLS/ELS – v-p-r Model	ELS with v, p and r Bias Enabled
RLS/ELS Time Step	0.2 s
Exponential Forgetting	0.99 for v-p-r Model
Directional Forgetting	On
Covariance Reset	Auto @ cond $P_{vpr} > 0.5 \times 10^6$, $P_{vpr} = P_{vpr} + 10*I$
Surge Model	Constant Surge Speed ($u[k] = u[k-1]$)

Table 5.3 Settings for Investigation of Sway–Roll–Yaw Models

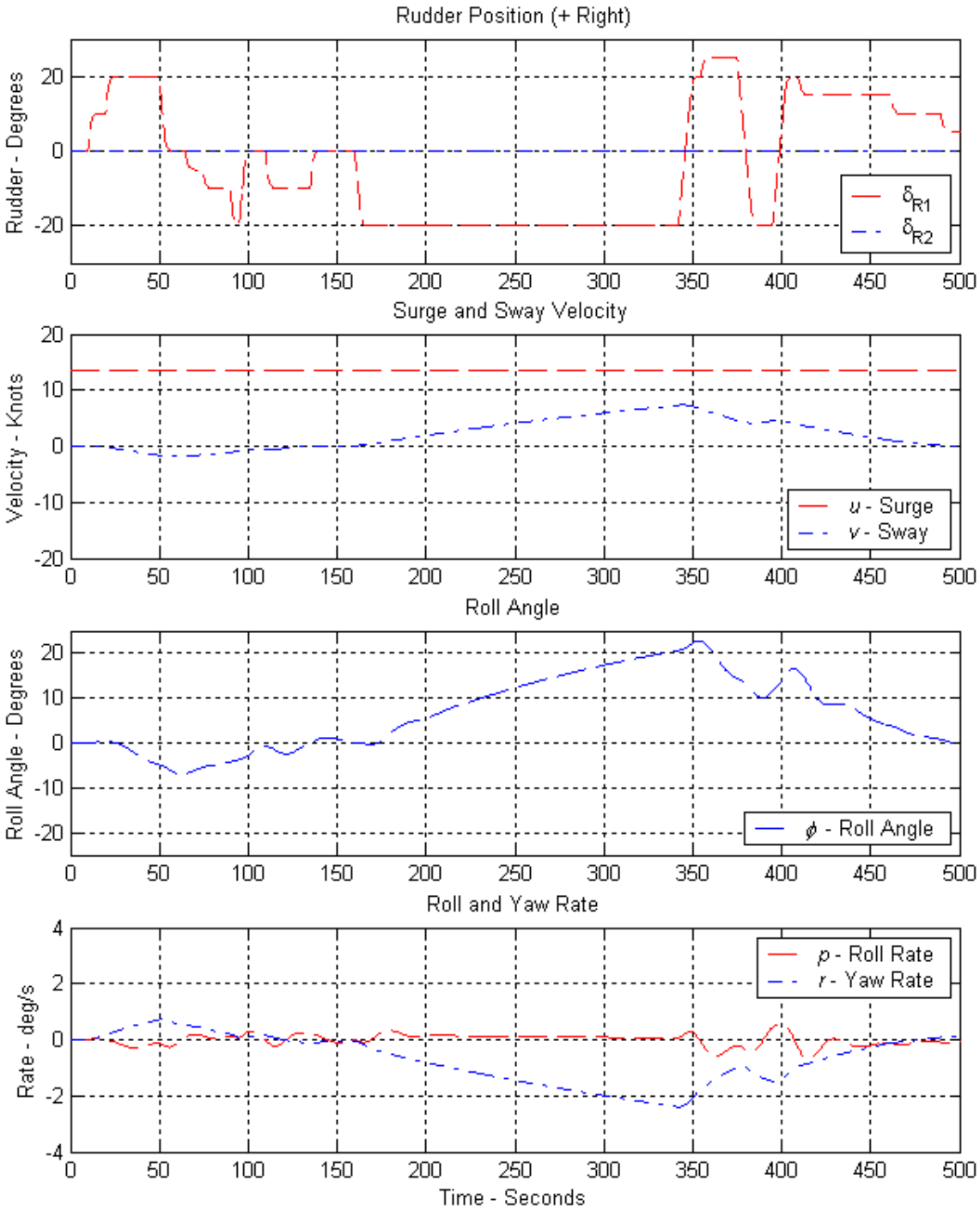


Figure 5.14 Vessel State Excitation for Sway–Roll–Yaw Model Comparison

1. Linear Yaw Rate Model

Parameter estimation and path prediction from the yaw rate model was compared to Fossen's linearized sway–roll–yaw containership model. The truth model control inputs and excited states are displayed in Figure 5.14. Yaw model parameter estimation versus time is displayed in Figure 5.15. The corresponding truth model parameter is plotted as a straight line for comparison. The control parameter term, yaw rate due to rudder deflection, was reasonably steady and close to the true parameter. This indicates that the yaw rate model adequately captures the steady–state yaw rate. However, the plant parameter, a'_{33}, showed poor parameter stability. It even became unstable passing 475 seconds. Figure 5.16 compares the estimated yaw model eigenvalue to the truth model sway and yaw rate eigenvalues. It is evident that the yaw rate model picked up influence from both the sway velocity and yaw rate. However, model adaptation lags control excitation. As result, the yaw rate model provided poor path prediction in maneuvering situations.

Figure 5.17 compares yaw rate model path prediction snapshots to the full truth model at time equal to 170, 200 and 230 seconds, respectively. The t = 170 seconds snapshot is shortly after initiation of a steady 20 degree left rudder input. The author immediately noted that path integration using initial vessel heading created poor results. This was evidenced by the immediate divergence between actual vessel path and predicted path when the vessel has sideslip. For the yaw rate model, the path integration initial condition needs to take into account vessel sideslip. As a result, the algorithm was modified to start with the vessel course through the water. Figure 5.18 displays the same snapshot comparison as Figure 5.17, but with the modified path integration initial condition. Path prediction is much better, but the yaw rate model still performed poorly. At 170 seconds, the yaw rate considerably over predicts turn rate and under predicts turn diameter. At 200 seconds, the opposite occurs. Finally, at 230 seconds (over one minute after the turn is commenced), the yaw rate model performs adequately. This confirms that the yaw rate model is inadequate unless a near–steady–state condition is achieved. This qualitatively confirms the conclusions from parameter and eigenvalue the observations.

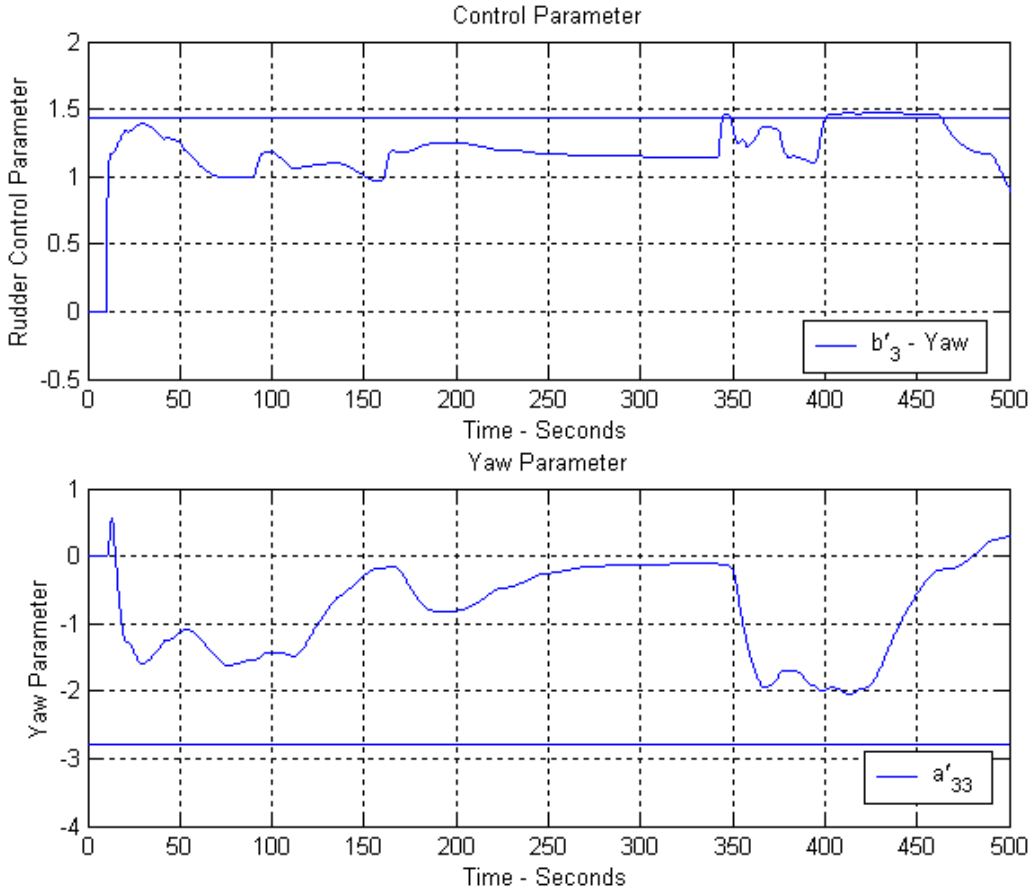


Figure 5.15 Yaw Rate Model Parameters Compared to Corresponding Linear Containership Truth Model Parameter ($\lambda = 0.99$)

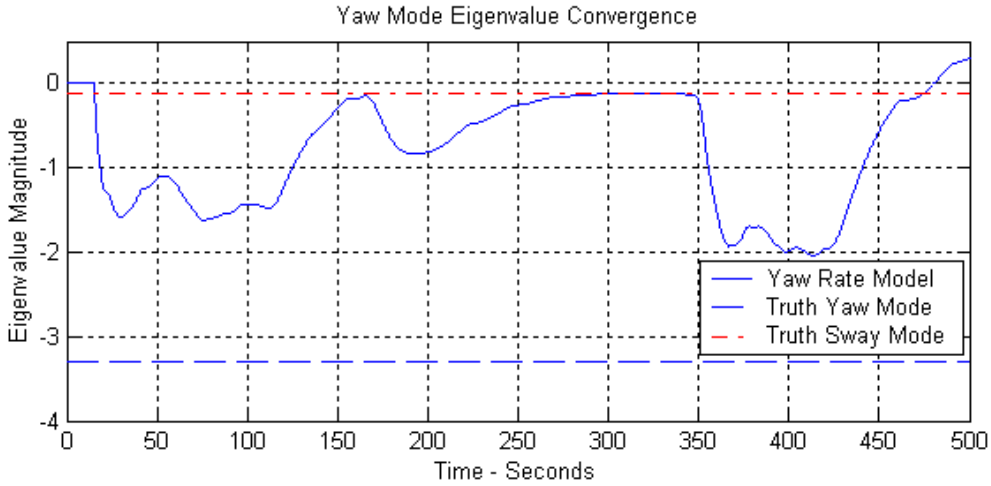
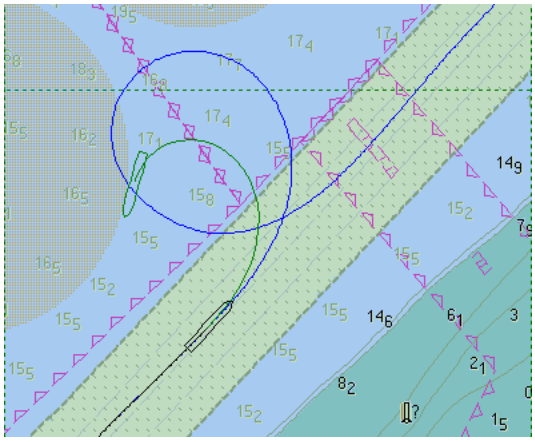
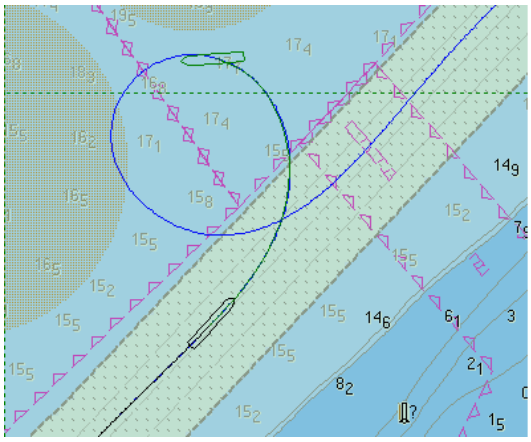


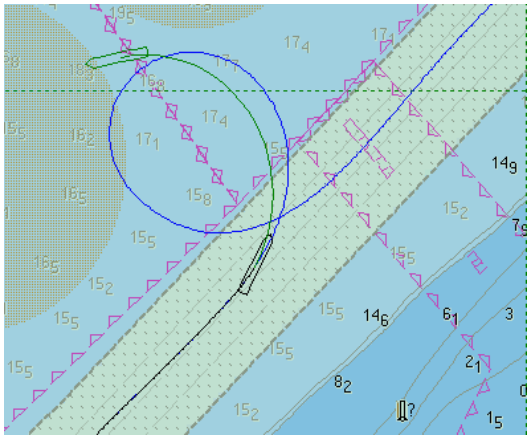
Figure 5.16 Yaw Rate Model Eigenvalue Compared to Truth Model Sway and Yaw Eigenvalues ($\lambda = 0.99$)



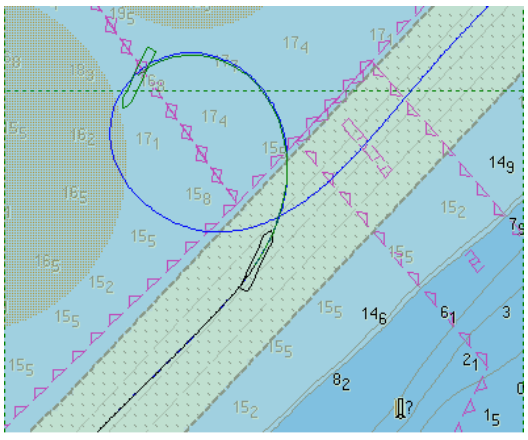
Yaw Rate Model (170 Seconds)



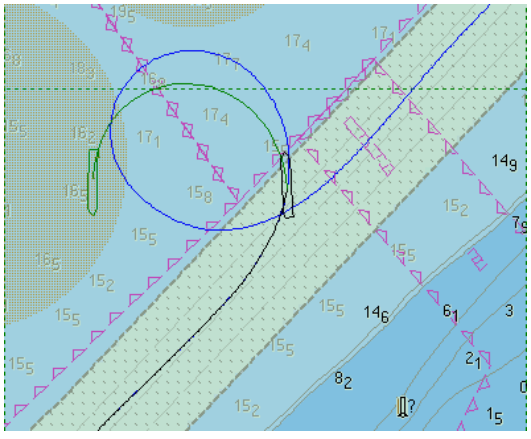
Linear Truth Model (170 Seconds)



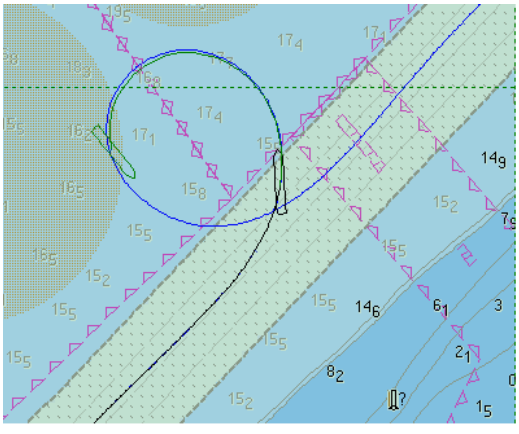
Yaw Rate Model (200 Seconds)



Linear Truth Model (200 Seconds)

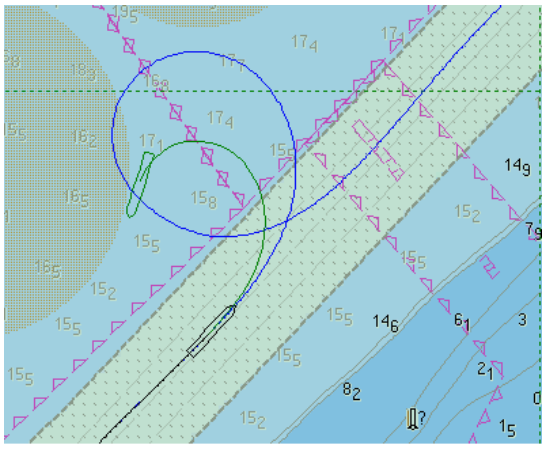


Yaw Rate Model (230 Seconds)

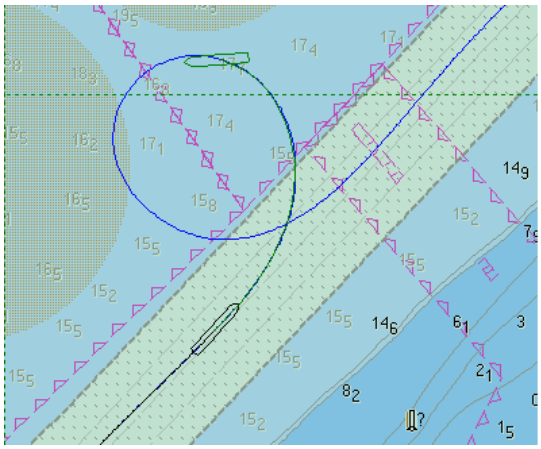


Linear Truth Model (230 Seconds)

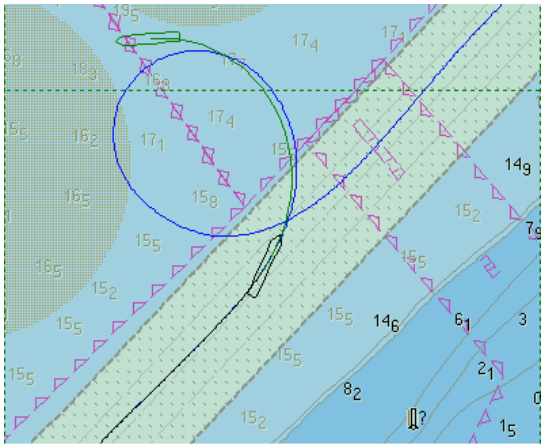
Figure 5.17 Yaw Rate Model Two Minute Path Prediction Compared to Truth Model Following 20 Degree Left Rudder Command



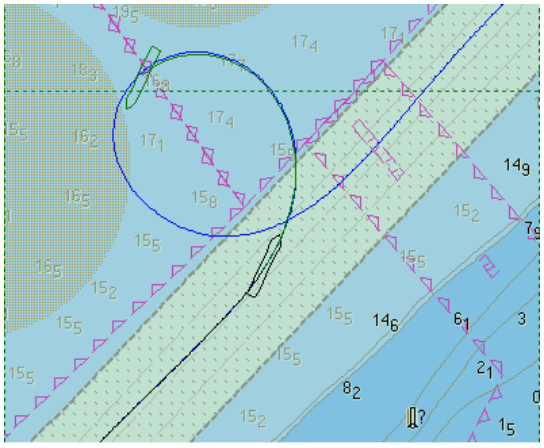
Yaw Rate Model (170 Seconds)



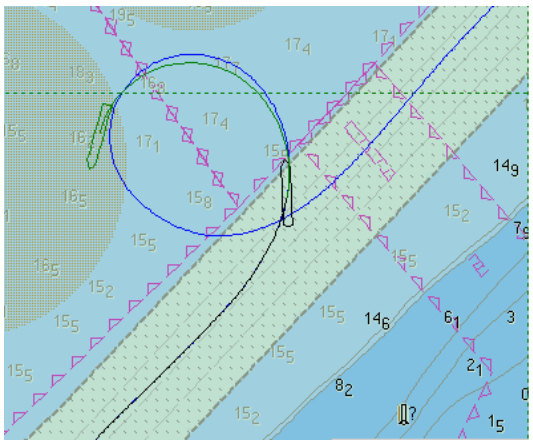
Linear Truth Model (170 Seconds)



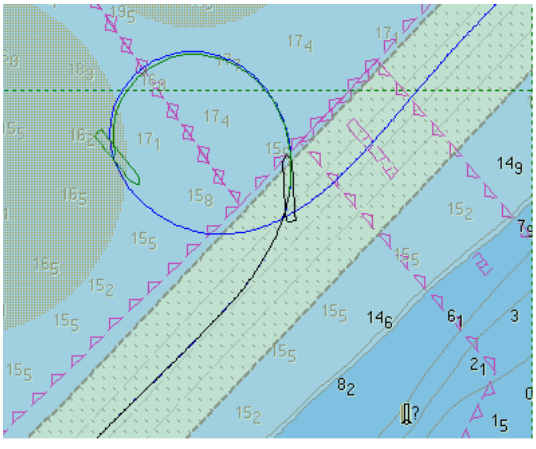
Yaw Rate Model (200 Seconds)



Linear Truth Model (200 Seconds)



Yaw Rate Model (230 Seconds)



Linear Truth Model (230 Seconds)

Figure 5.18 Yaw Rate Model Path Prediction with Modified Initial Condition Compared to Truth Model Following a 20 Degree Left Rudder Command

2. Linear Sway–Yaw Model

Parameter estimation and path prediction from the sway–yaw model was compared to Fossen's linearized sway–roll–yaw containership model. The same truth model control inputs and excited states (Figure 5.14) were provided to the sway–yaw model as the yaw rate only model. Sway–yaw model parameter estimation versus time is displayed in Figure 5.19. The corresponding estimated control parameters were accurate and reasonably steady up to the series of rapid rudder shifts around 340 seconds. The sway–yaw plant parameters were also reasonably steady up until the rapid control inputs. However, the parameters tended to be faster than their truth model counterparts. This is due to the sway–yaw model capturing motion input from the faster truth model roll mode.

The estimated sway–yaw model eigenvalues are compared to truth model eigenvalues in Figure 5.20. The eigenvalues were fairly steady up until the high bandwidth rudder reversal. The eigenvalues are somewhat faster than the truth model counterparts due to capturing of the faster roll mode influence.

The steady nature of the predicted eigenvalues at low to medium control bandwidths makes this model suitable for path prediction in all but the most dynamic maneuvering situations.

Figure 5.21 compares sway–yaw model path prediction snapshots to the full truth model at time equal to 170, 200 and 230 seconds, respectively. The t = 170 seconds snapshot is shortly after commencing a 20 degree left rudder turn. Path prediction is nearly identical to that obtained from the higher order truth model. Clearly, sway–yaw model fidelity is sufficient to capture the majority of vessel motion for depiction of future vessel path. However, a higher order sway–roll–yaw model is required if roll angle estimation is required.

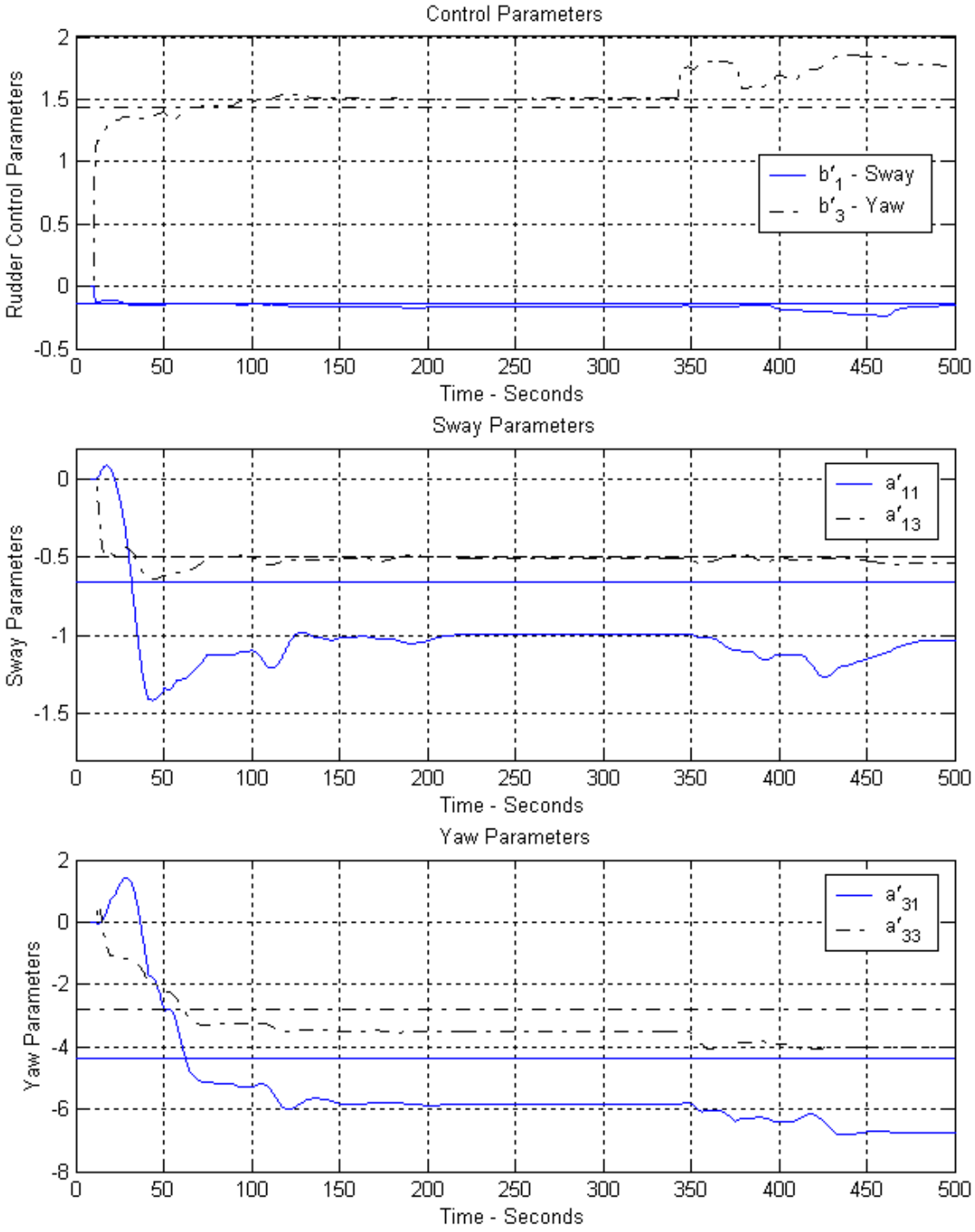


Figure 5.19 Sway–Yaw Model Parameters Compared to Corresponding Linear Containership Truth Model Parameters ($\lambda = 0.99$)

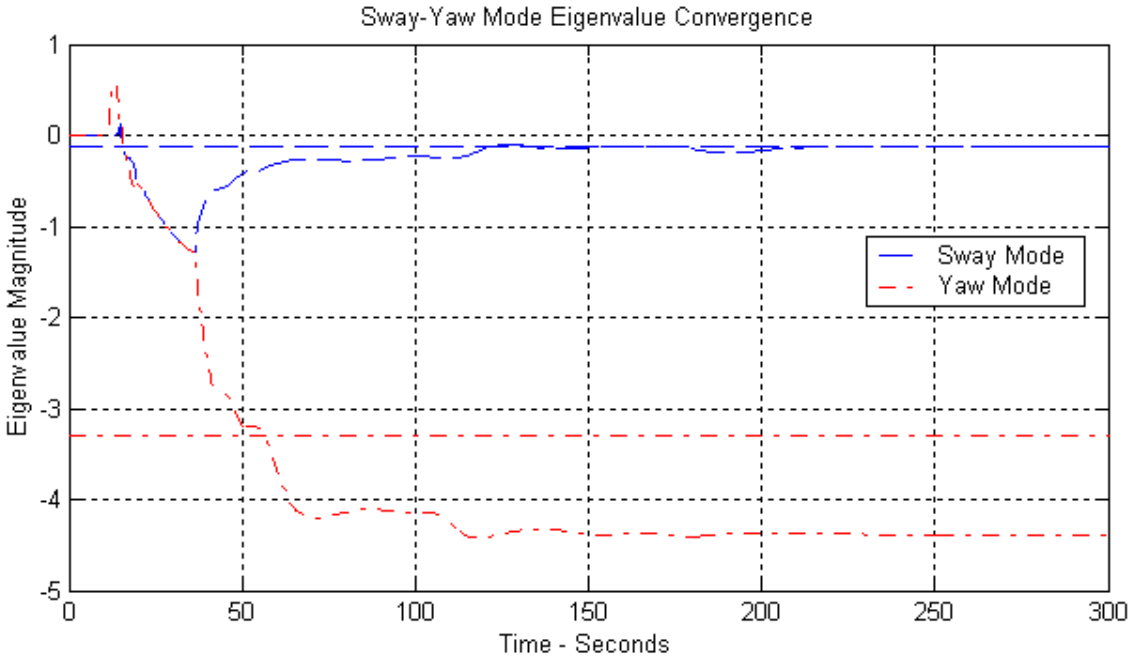
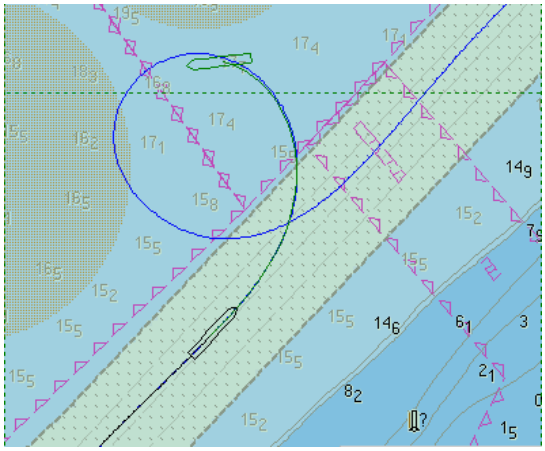
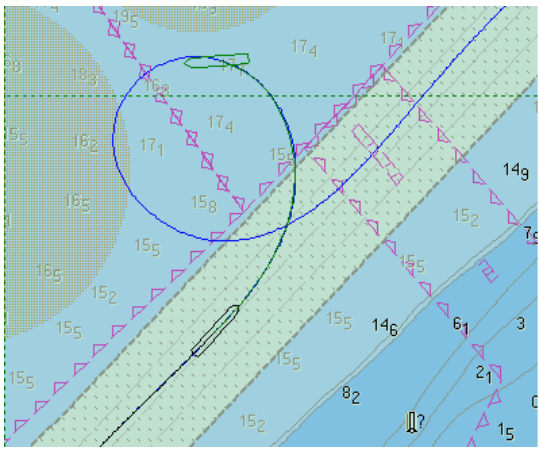


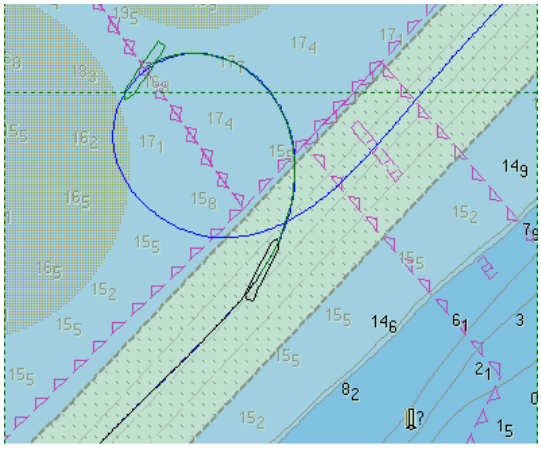
Figure 5.20 Sway-Yaw Model Eigenvalues Compared to Truth Model Sway and Yaw Eigenvalues



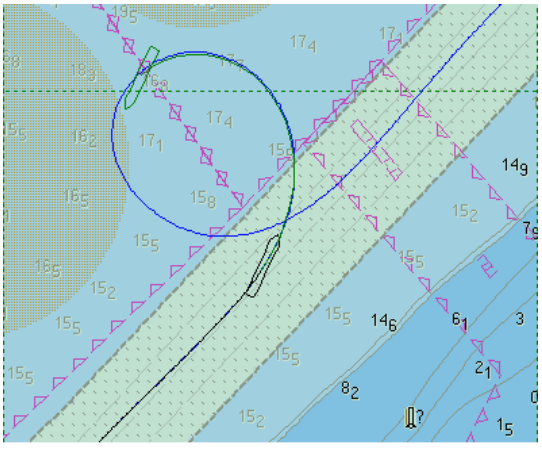
Sway–Yaw Model (170 Seconds)



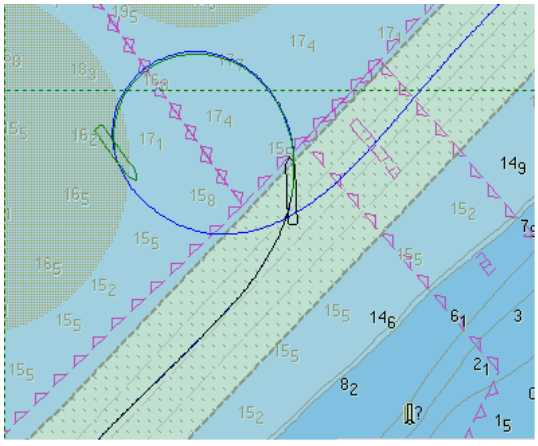
Linear Truth Model (170 Seconds)



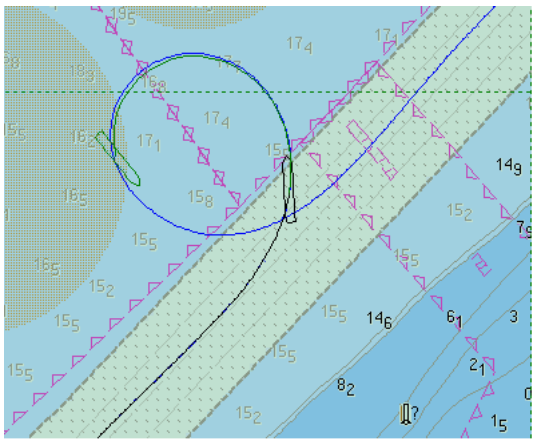
Sway–Yaw Model (200 Seconds)



Linear Truth Model (200 Seconds)



Sway–Yaw Model (230 Seconds)



Linear Truth Model (230 Seconds)

Figure 5.21 Sway–Yaw Rate Model Two Minute Path Prediction Compared to Truth Model Following a 20–Degree Left Rudder Command

D. INVESTIGATION OF LINEAR MODEL MATCHING TO A NON-LINEAR TRUTH MODEL

Section C verified the suitability of both the sway–yaw and the sway–roll–yaw models for path prediction. This section will extend those results to an evaluation of how well these linear motion models path predict despite underlying non-linear vessel motion. The non-linear Fossen containership model is used for truth data.

The same series of control inputs is applied to the non-linear truth model as with the linear model used in Section C. However, a lower level of excitation and different vessel path is obtained with the non-linear model. In general, the non-linear model response is lower in all the states at high control input levels. Exponential forgetting was enabled in an attempt to capture the non-linear motion through linear model adaptation. The system was initialized to the settings in Table 5.4. Figure 5.22 displays the control inputs and states that stimulated the linear sway–yaw and sway–roll–yaw parameter estimation models.

Parameter	***Setting***
Truth Data	Non-Linear Containership Model – Uo = 7 m/s
Truth Model Time Step	0.2 s (RK2 Integration)
RLS/ELS – *v-p-r* Model	ELS with *v*, *p* and *r* Bias Enabled
RLS/ELS Time Step	0.2 s
Exponential Forgetting	0.99 for *v-p-r* Model
Directional Forgetting	On
Covariance Reset	Auto @ cond $P_{vpr} > 0.5 \times 10^6$, $P_{vpr} = P_{vpr} + 10*I$
Surge Model	Non-Linear Surge ($\lambda_u = 0.995$)

Table 5.4 Settings for Investigation of Linear Model Matching to a Non-linear Truth Model

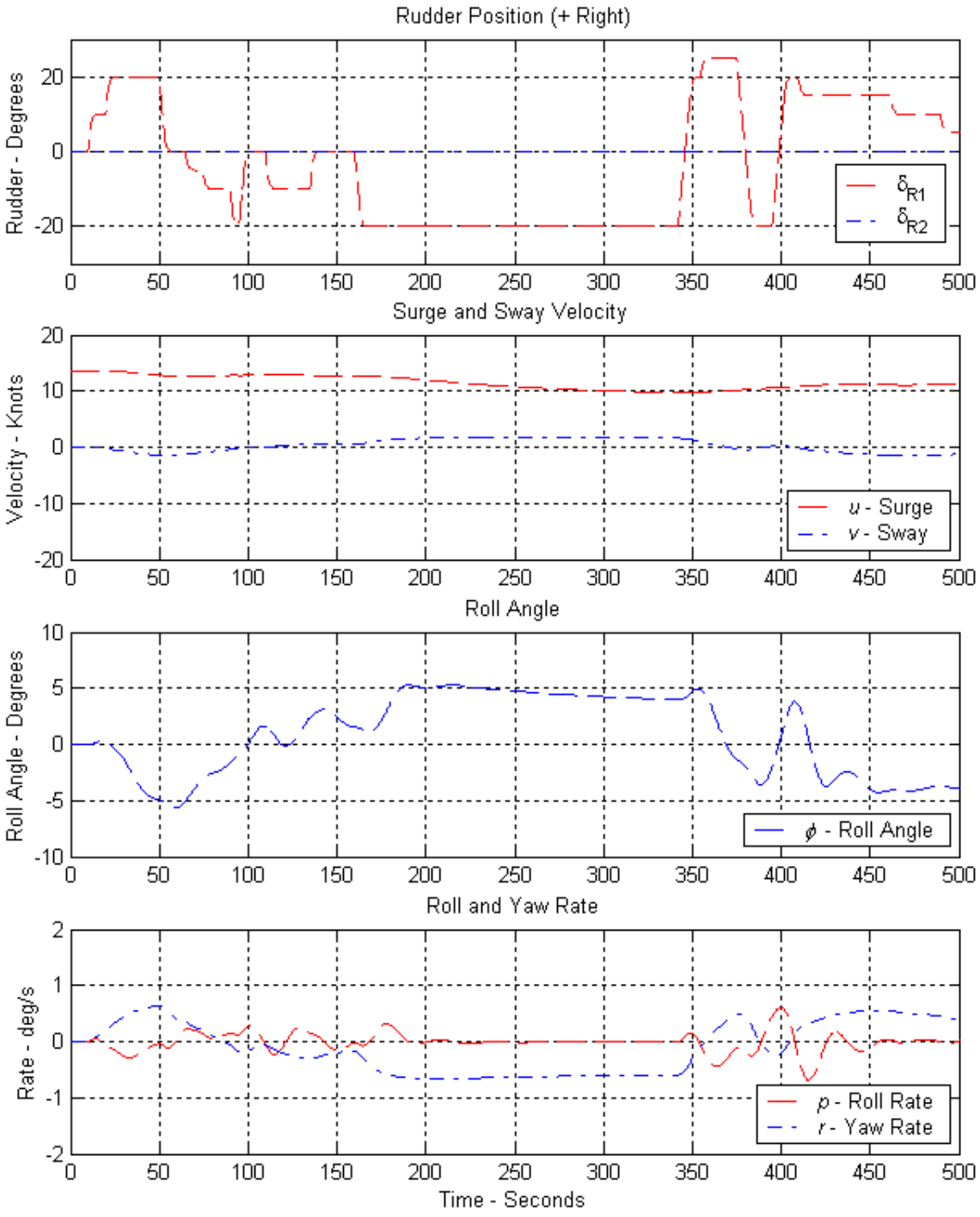


Figure 5.22 Vessel State Excitation for Investigation of Linear Model Matching to a Non-linear Truth Model

1. Linear Sway–Yaw Model Parameter Estimation from a Non-linear Truth Model

Parameter estimation and path prediction from the sway–yaw model was compared to Fossen's non-linear surge–sway–roll-yaw containership model. The control inputs and excited states are displayed in Figure 5.22. Sway–yaw model parameter estimation versus time is displayed in Figure 5.23 along with the non-linear model parameters linearized at U_o = 7 m/s. The control parameters remain reasonably steady throughout the maneuvers, indicating probable good performance under steady–state conditions. The sway and yaw plant parameters display more variation. This indicates the exponential forgetting is attempting to adjust the linear model to the non-linear dynamics. This is quite evident shortly after a steady 20–degree left rudder input is executed at 265 seconds. Most parameters require approximately 60 seconds to adjust to the new set of parameters for a steady 20–degree left rudder turn. The parameters adjust again following a series of rapid rudder shifts around 340 seconds.

The sixty–second adaptation window is reasonable based on the selected exponential forgetting time constant. Table 3.1 indicates an exponential time constant of 20 seconds given λ = 0.99 and Δ = 0.2 sec. Allowing three time constants to reach steady–state fits well with the observed 60-second parameter adaptation time. The exponential forgetting factor will need to be reduced if faster parameter adjustment is desired. The obvious fall out of increasing forgetting (smaller λ) is higher susceptibility to noise.

Path prediction snapshots following initiation of a sustained 20–degree left rudder turn are presented in Figure 5.25. The path prediction snapshots are overlaid over the actual non-linear truth model path. The path prediction at 170 seconds overestimates the turn performance. The prediction at 200 seconds is slightly improved. The prediction at 230 seconds is much improved and nicely overlays the actual path. Roughly 60 seconds are required for the ELS algorithm to pick up the non-linear parameter shift. This qualitative observation supports the expectations based on the parameter estimation behavior observed. A sixty–second adjustment interval is relatively slow. Especially when one recognizes that the model will take 60 seconds to adjust back to the linear parameters around the equilibrium point. As a result, the useful bandwidth of the path prediction al-

gorithm is relatively narrow. A reduced forgetting factor or addition of some non-linearity in the model may produce improved results.

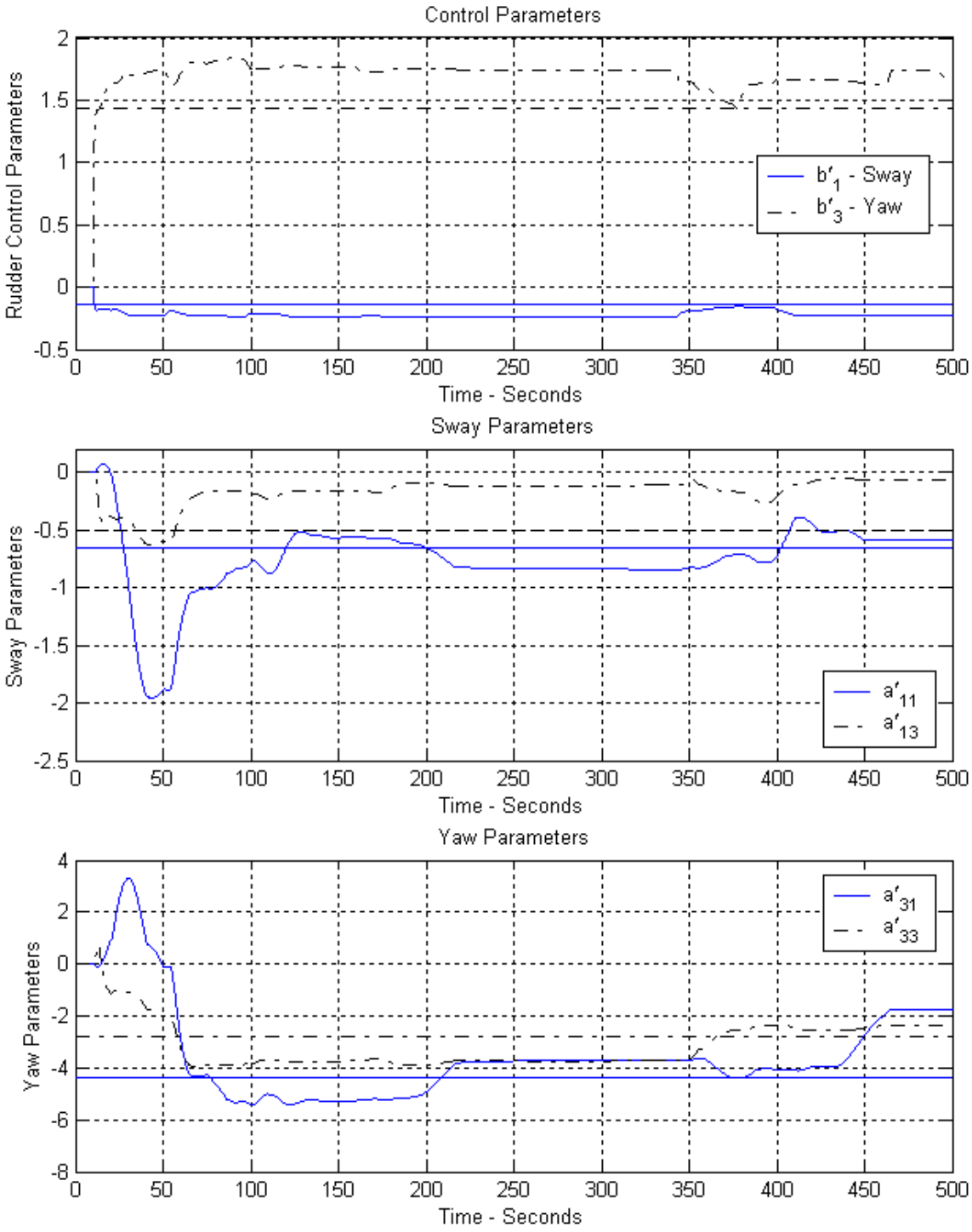


Figure 5.23 Sway–Yaw Model Parameters Compared to Non-Linear Containership Parameters Linearized Around U_o = 7 m/s (λ = 0.99)

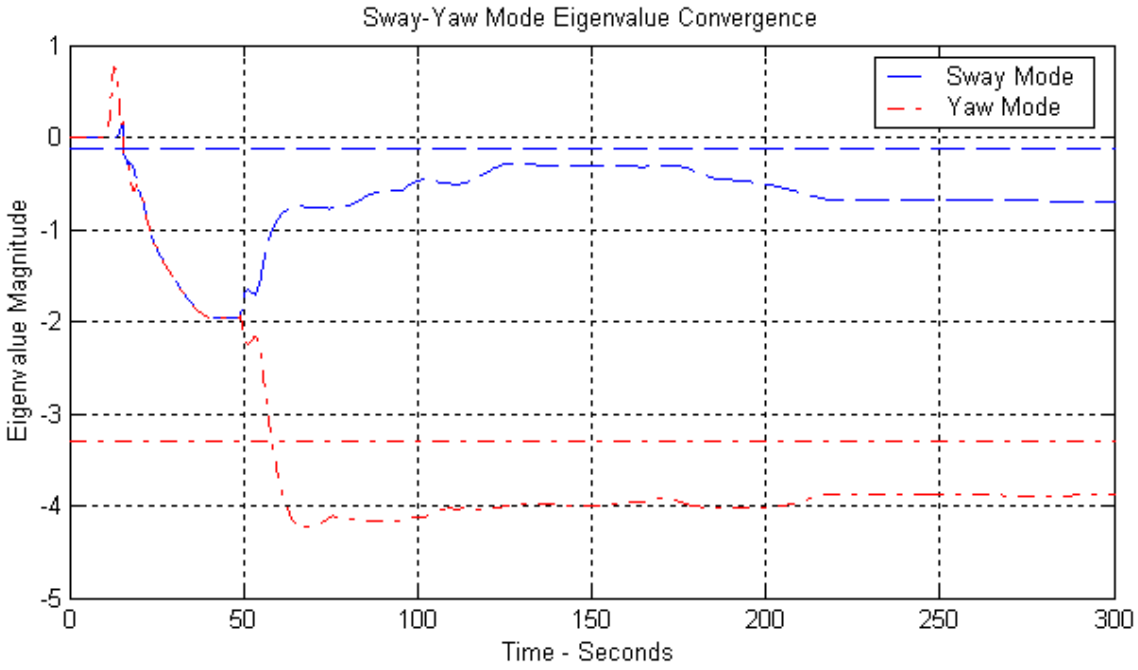
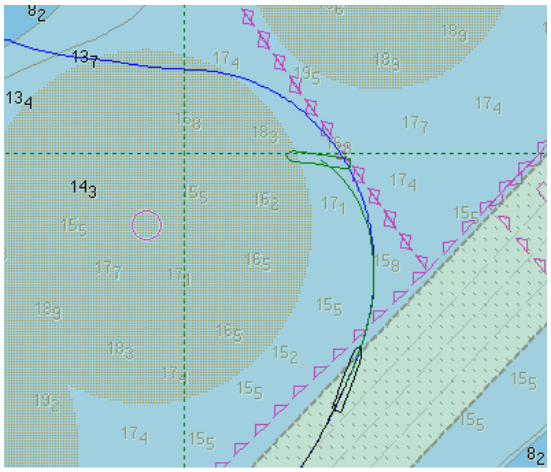
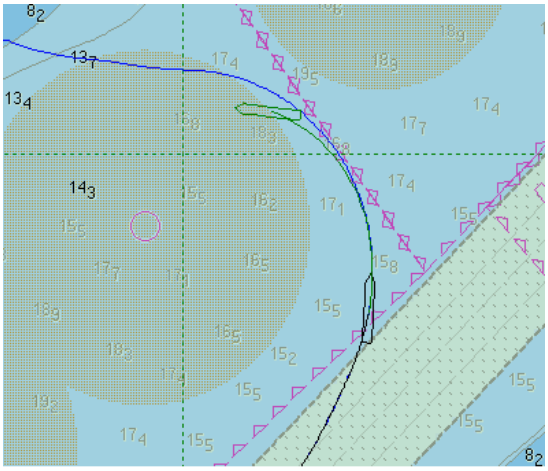


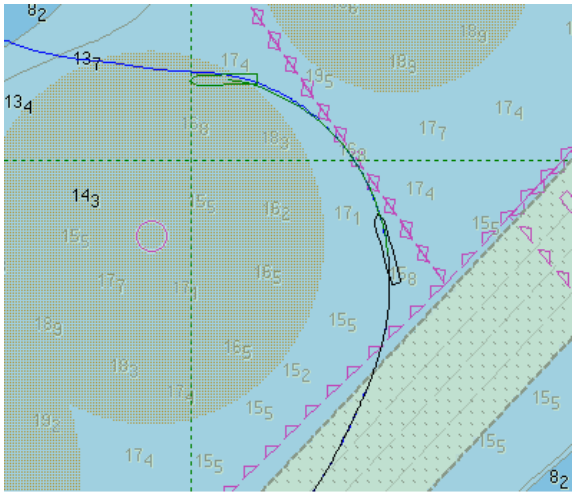
Figure 5.24 Sway–Yaw Model Eigenvalues Compared to Non-Linear Containership Eigenvalues Linearized Around U_o = 7 m/s (λ = 0.99)



Linear Sway–Yaw Model (170 sec)



Linear Sway–Yaw Model (200 sec)



Linear Sway–Yaw Model (230 sec)

Figure 5.25 Sway–Yaw Model Two Minute Path Prediction Compared to Truth Model Path Following a Steady 20–Degree Left Rudder Command

2. Linear Sway–Roll–Yaw Model Parameter Estimation from a Non-linear Truth Model

Parameter estimation and path prediction from the sway–roll–yaw model was compared to Fossen's non-linear surge–sway–roll–yaw containership model. The control inputs and excited states are displayed in Figure 5.22. Sway–roll–yaw model parameter estimation versus time is displayed in Figure 5.26 along with the non-linear model parameters linearized around U_o = 7 m/s. As with the sway–yaw model, the control parameters remain reasonably steady throughout the maneuvers, indicating probable good performance under steady–state conditions. The one exception is the roll control parameter. A large fluctuation is evident during the rapid rudder reversals starting at 340 seconds. The sway, roll and yaw plant parameters display more variation than with the sway–yaw model. This is expected when there are 15 degrees of freedom (parameters) in the sway–roll–yaw model versus six for the sway–yaw model. Despite the greater variations in parameters, the eigenvalues remain well behaved. The exponential forgetting factor will need to be reduced if faster parameter adjustment is desired. The obvious fall out of increasing forgetting (smaller λ) is higher susceptibility to noise.

Path prediction snapshots for the sway–roll–yaw model are presented in Figure 5.27 for a sixty–second period following initiation of a 20–degree left rudder input. The estimated path is overlaid on the actual path generated with the non-linear truth model. Results are nearly identical to the sway–yaw model. Path prediction at 170 seconds overestimates the turn performance. The prediction at 200 seconds is slightly improved. The prediction at 230 seconds is much improved and nicely overlays the actual path. Roughly 60 seconds are required for the ELS algorithm to pick up the non-linear parameter shift. As with the sway–yaw model, a sixty second adjustment interval is relatively slow, especially when considering another 60 seconds is required to adjust back to the parameters close to equilibrium. As a result, the useful bandwidth of the linear path prediction models is relatively narrow. A reduced forgetting factor or addition of some non-linearity in the model may produce faster results.

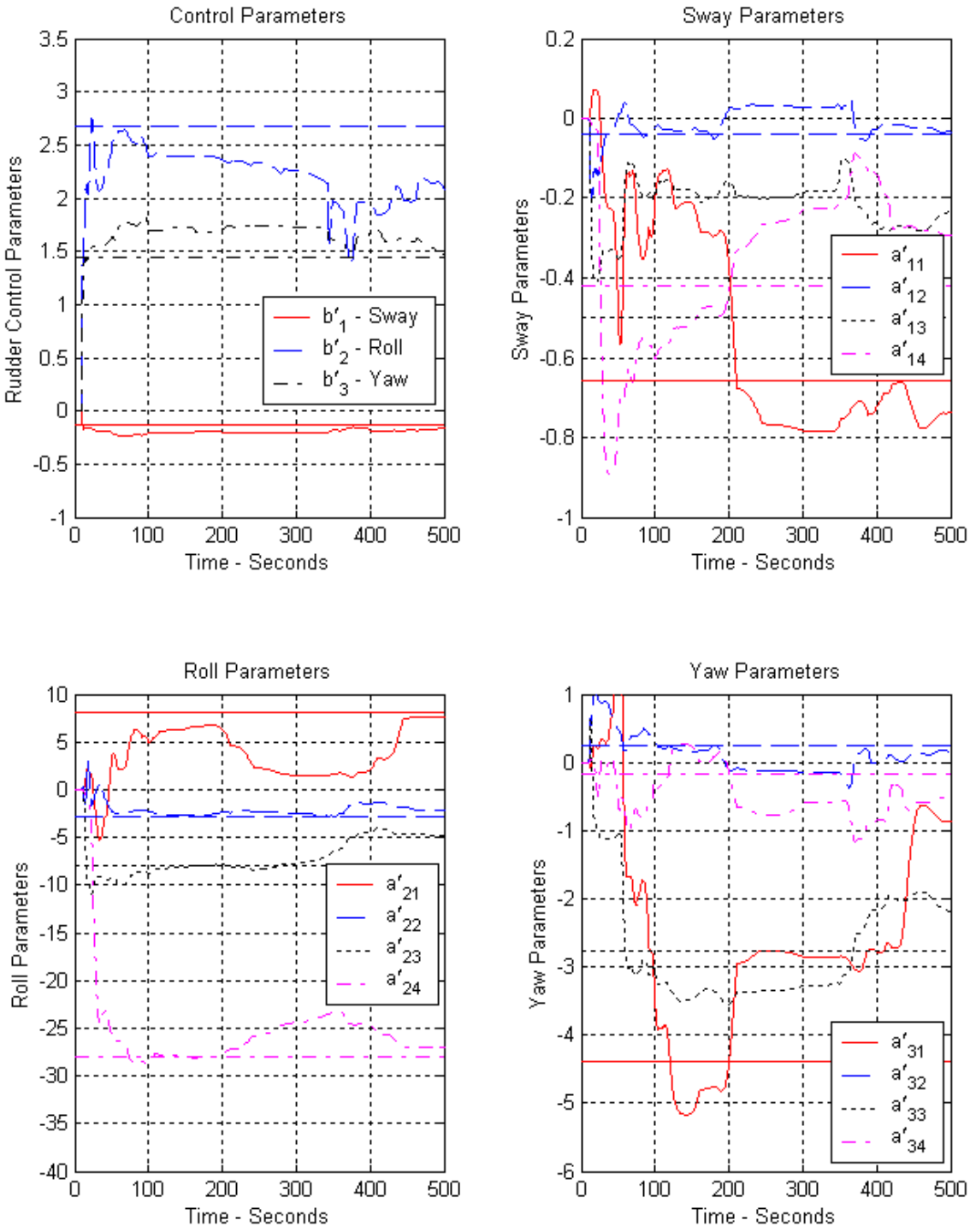


Figure 5.26 Sway–Roll–Yaw Model Parameters Compared to Non-Linear Container-ship Parameters Linearized Around U_o = 7 m/s (λ = 0.99)

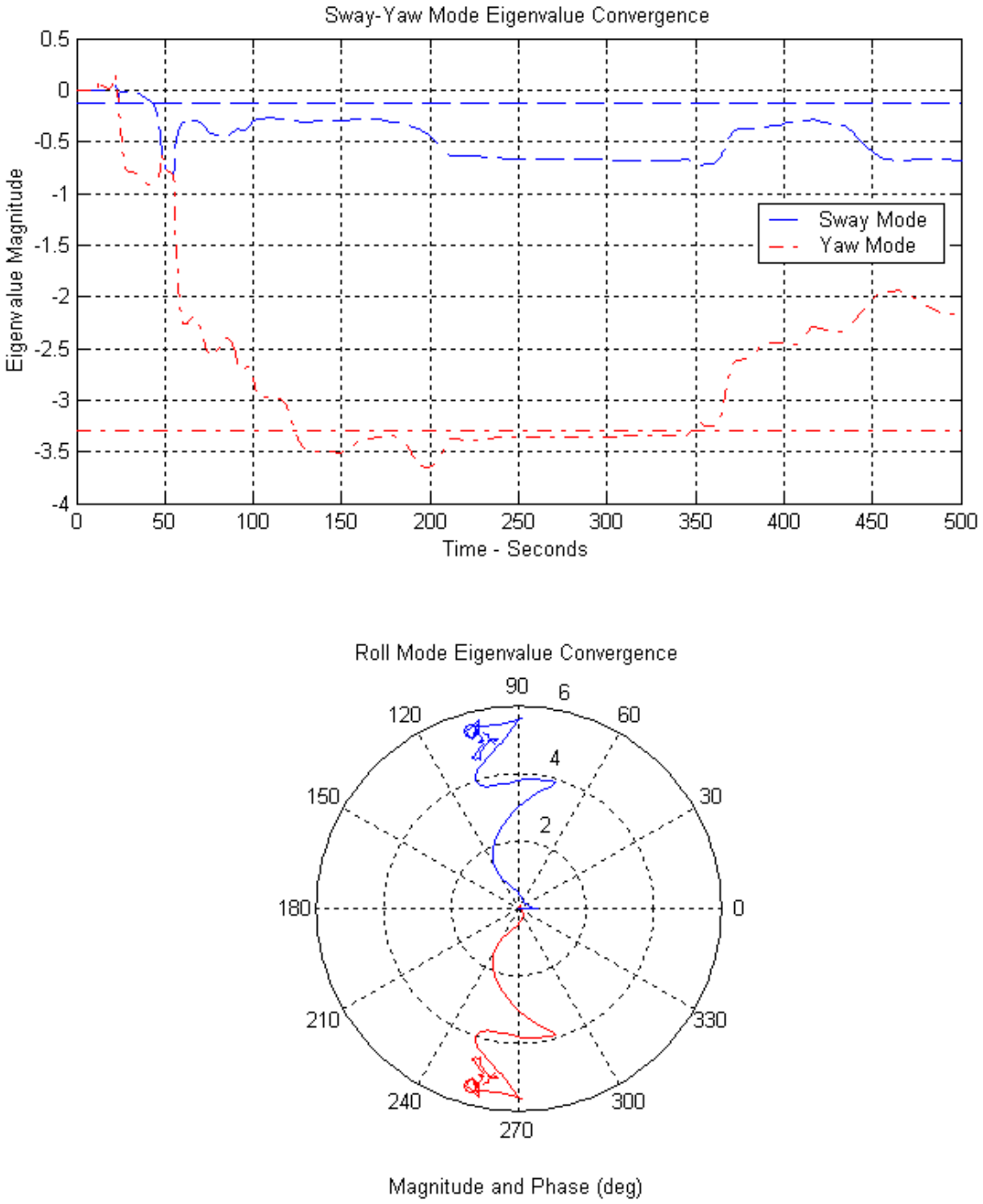
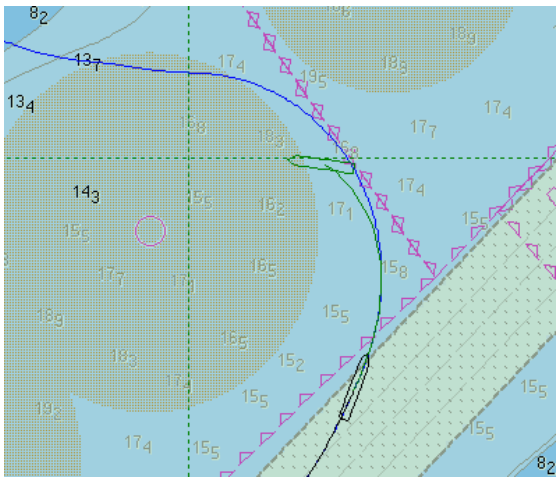
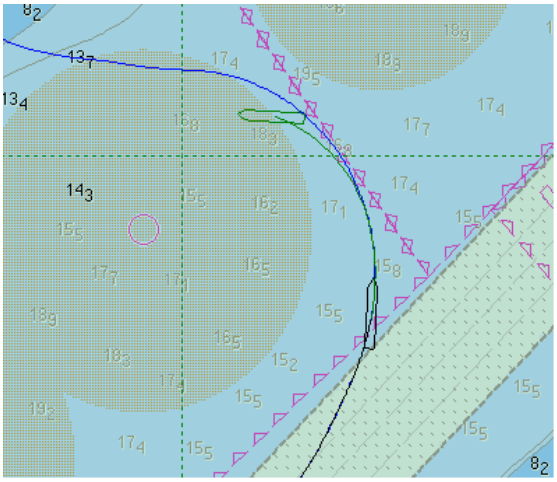


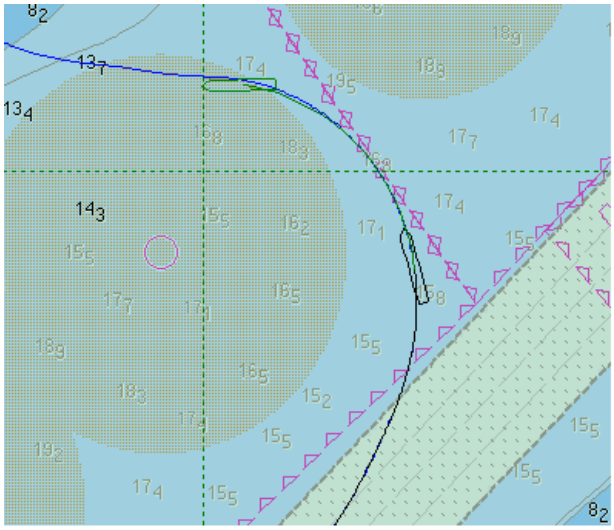
Figure 5.27 Sway–Roll–Yaw Model Eigenvalues Compared to Non-Linear Container-ship Eigenvalues Linearized Around U_o = 7 m/s



Linear Sway–Roll–Yaw Model (170 sec)



Linear Sway–Roll–Yaw Model (200 sec)



Linear Sway–Roll–Yaw Model (230 sec)

Figure 5.28 Sway–Roll–Yaw Model Two Minute Path Prediction Compared to Truth Model Path Following a Steady 20 Degree Left Rudder Command

E. VERIFICATION OF CURRENT AND SENSOR BIAS ESTIMATION

This section verifies the correct estimation of current and sensor biases. Sensor biases observability is limited to roll angle and rudder angle. Rate sensor biases are not observable. However, this is not a limitation provided that the rate data is obtained from an accurate inertial platform. Differentiating angle measurements will also generate unbiased data. In general, the estimation algorithms worked well. Sensor bias estimates drifted during high bandwidth control inputs due to inaccurate parameter estimates. However, the overall system maintained accurate path prediction despite the drifting estimates of the biases. A standard series of rudder input excitation was applied to linearized sway–roll–yaw model for approximately 500 seconds. Exponential forgetting was enabled. The system was initialized to the settings in Table 5.5 and Figure 5.29 displays the control inputs and states from the linearized containership truth model.

Parameter	***Setting***
Truth Data	Linearized Containership Model – Uo = 7 m/s
Truth Model Time Step	0.2 s (RK2 Integration)
RLS/ELS – *v-p-r* Model	ELS with *v*, *p* and *r* Bias Enabled
RLS/ELS Time Step	0.2 s
Exponential Forgetting	0.99 for *v-p-r* Model
Directional Forgetting	On
Covariance Reset	Auto @ cond $P_{vpr} > 0.5 \times 10^6$, $P_{vpr} = P_{vpr} + 10*I$
Surge Model	Constant Surge Speed ($u[k] = u[k-1]$)
Roll Angle Bias	-2. °
Rudder Angle Bias	1.5°
Current	Set: 290° @ 1.5 knots

Table 5.5 Settings for Investigation of Current and Sensor Biases

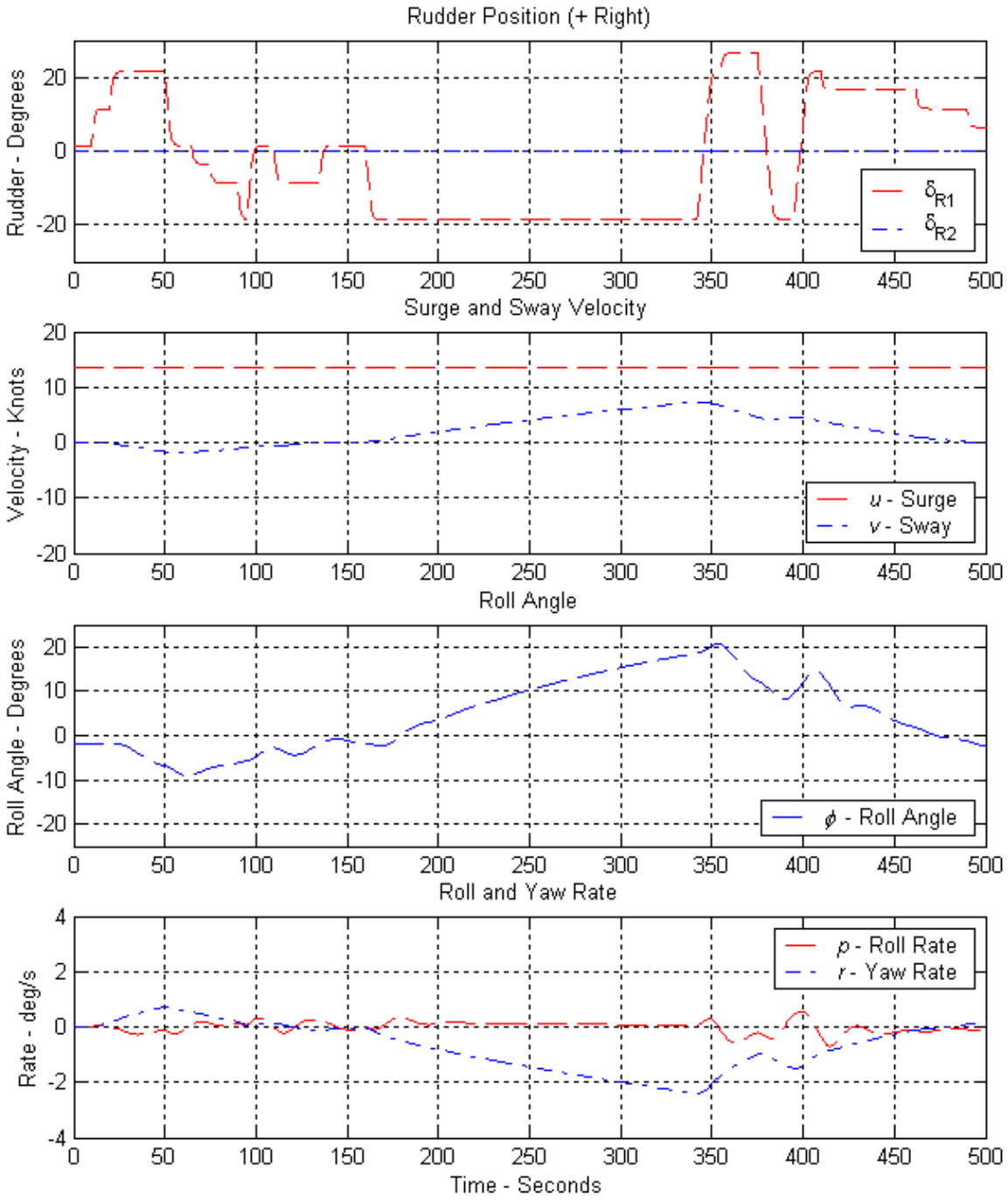


Figure 5.29 Vessel States for Verification of Current and Sensor Biases

1. Verification of Current Estimation

Current is estimated with a simple moving average filter. The difference between inertial and through–the–water velocity is averaged across five measurements. Measurements are restricted to low rudder deflection and turn rate situations. This prevents feedback between current estimation and parameter estimation during dynamic maneuvers.

The linear sway–roll–yaw model was fed data generated from the linear containership model. Control inputs and states are displayed in Figure 5.29. A current of 290° set (direction current is going) and 1.5 knot drift (velocity) was injected into the simulation. Current was held constant through the test. The current filter immediately calculated a set of 290.02° and drift of 1.50 knots and remained there for the rest of the data run. This confirmed the current estimation algorithm in a noise–free environment. Figure 5.30 displays parameter estimation progress and Figure 5.31 displays eigenvalue estimation versus time. Both are similar to the no–current situation (Figures 5.8 and 5.9). This confirms that a constant current does not effect parameter estimation.

Snapshots of path prediction with current at 170, 200 and 230 seconds are displayed in Figure 5.32. The snapshots on the left are with current set to 290° and 1.50 knots. The set of snapshots on the right are from the identical no–current situation. The same commands were provided to both simulations. The lateral displacement of the actual track line and distortion of the 360° turn in the left hand series of snapshots confirm proper implementation of current in the simulation routine. The close overlay of the predicted and actual paths indicates proper current estimation and path prediction.

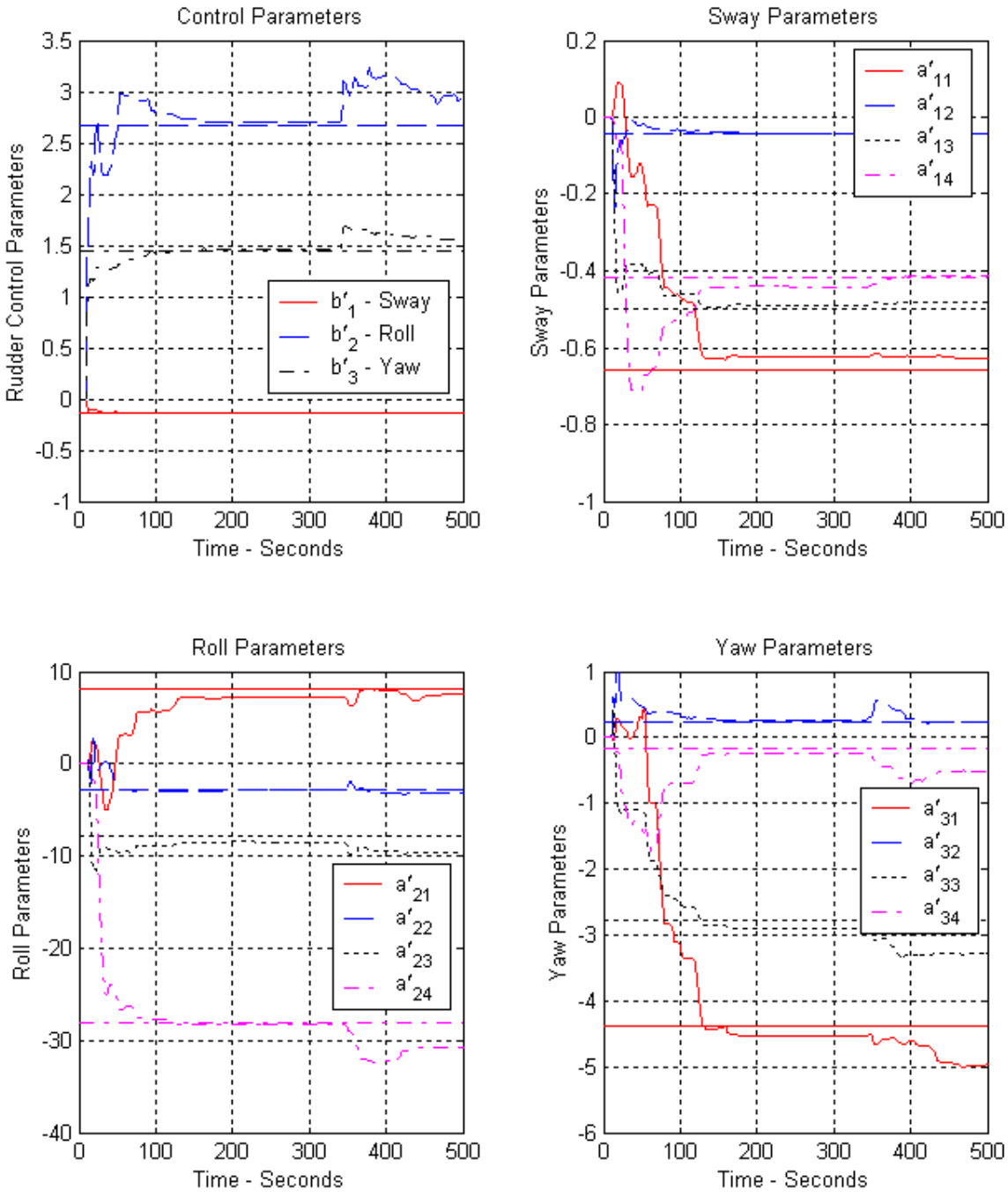


Figure 5.30 Sway–Roll–Yaw Parameter Estimation with Current Compared to Linear Containership Parameters (U_o = 7 m/s) (λ = 0.99)

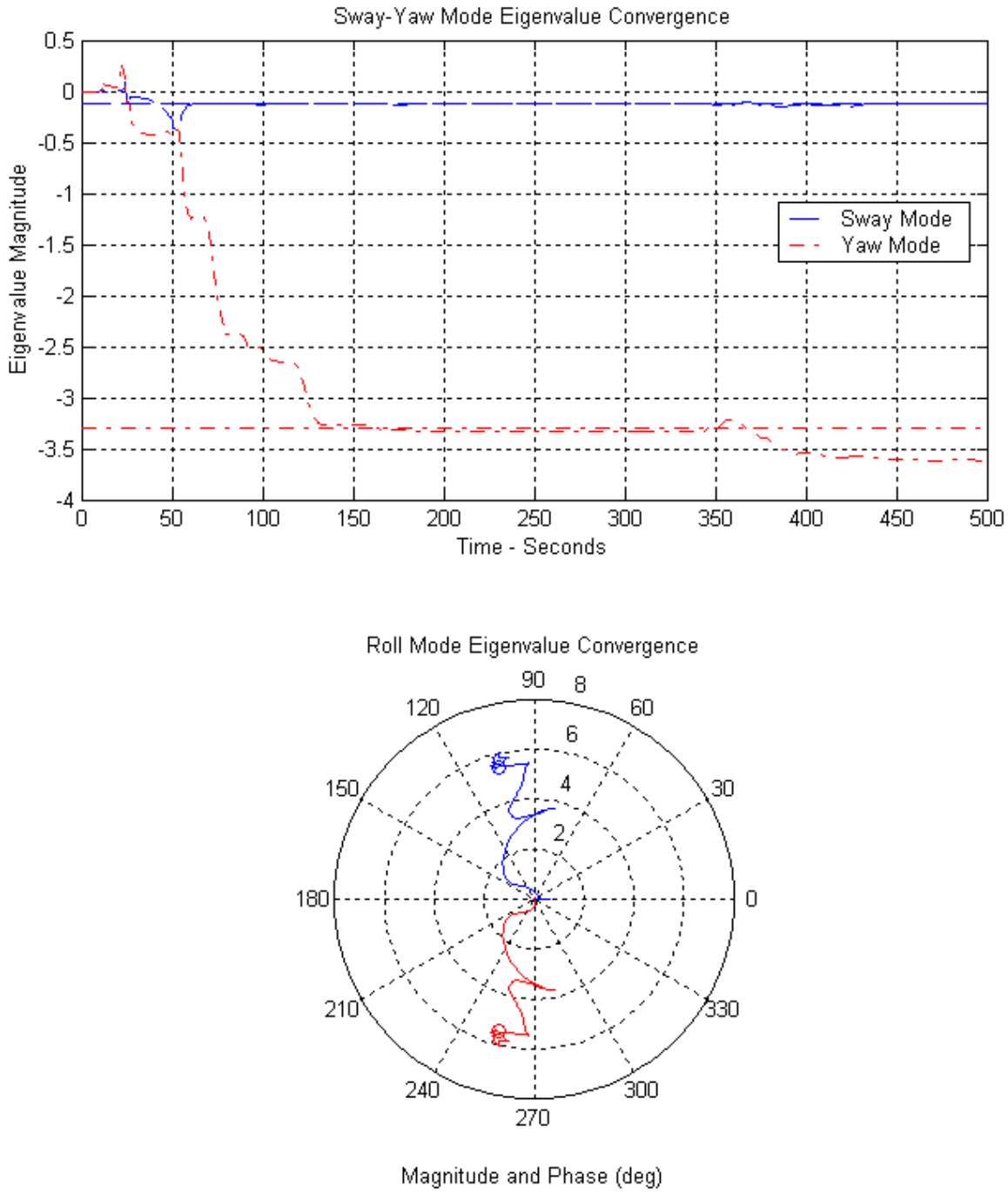
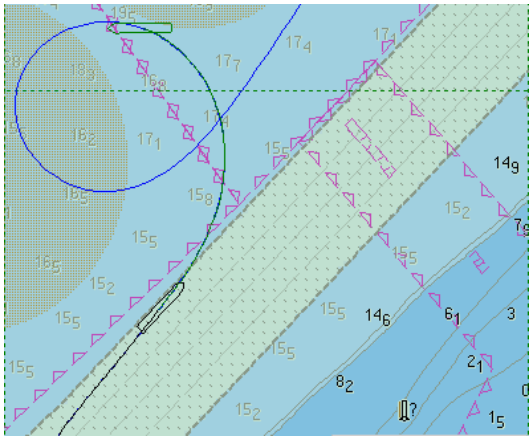
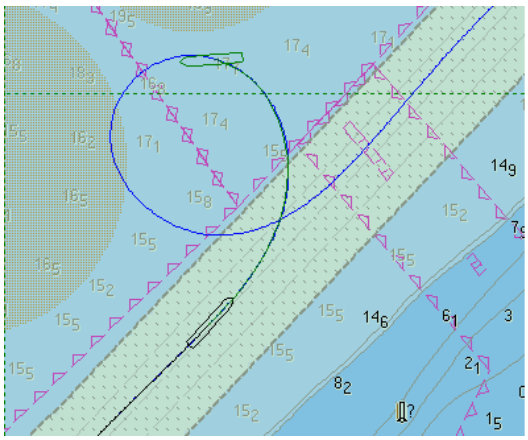


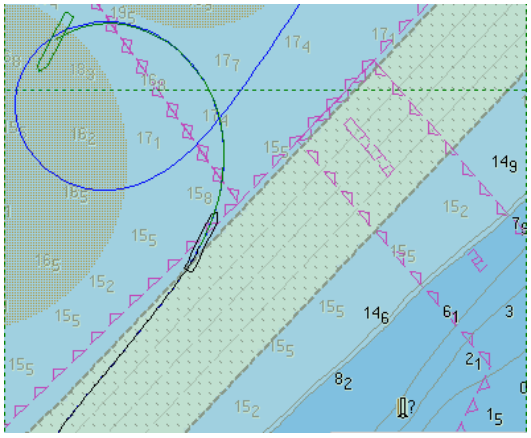
Figure 5.31 Sway–Roll–Yaw Eigenvalue Estimation with Current Compared to Linear Containership Eigenvalues (U_o = 7 m/s)



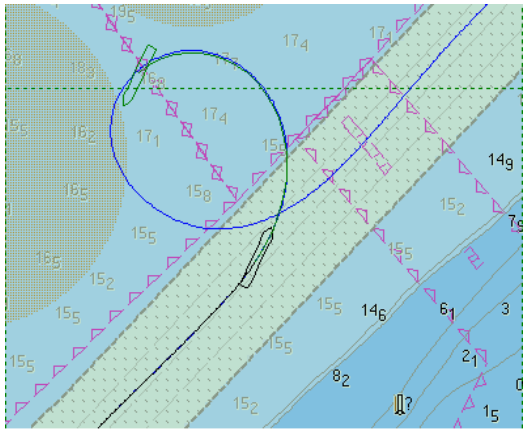
With Current – 290°/1.5 knt (170 sec)



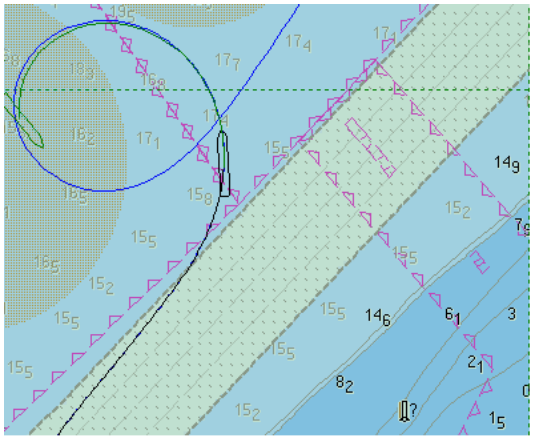
No Current Truth Model (170 sec)



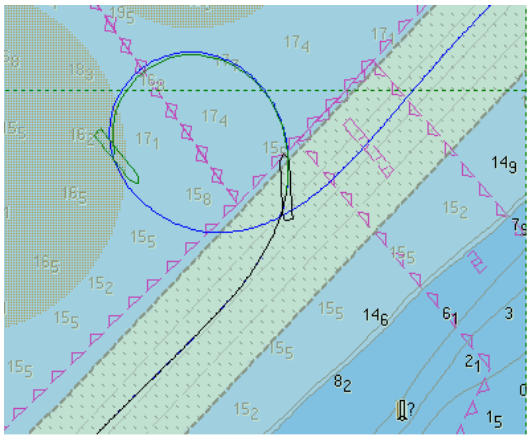
With Current – 290°/1.5 knt (200 sec)



No Current Truth Model (200 sec)



With Current – 290°/1.5 knt (230 sec)



No Current Truth Model (230 sec)

Figure 5.32 Path Prediction with Current Estimation Compared to Truth Model without Current (Constant 20–Degree Left Rudder Command)

2. Verification of Rudder and Roll Angle Bias Estimation

Rudder and roll angle error is modeled in the ELS algorithm as a slowly varying bias term. The sway, roll and yaw models each have a bias term which is an aggregate of roll and rudder angle measurement error and environmental disturbances. For this test, no environmental disturbances were introduced, leaving the biases a combination of rudder and roll angle error. Estimation of the individual biases is not important to path prediction. However, specific estimation of roll and rudder error is interesting for verification purposes. A few lines of code were added to the Parameter Estimation Module to isolate and output estimated roll and rudder angle bias. Combinations of the two biases were contained in three independent equations. Three equations and two unknowns creates an over specified situation, so a least–squares fit was selected.

Figure 5.33 displays the progress of bias estimation with an actual –2.0° roll and +1.5° rudder error. Initial convergence is reached by 130 seconds and both bias estimates are well behaved and accurate up until the high bandwidth rudder reversals commencing at 340 seconds. During the rapid rudder reversals, rudder bias estimation reaches +5° or a 3° error. The estimated roll angle bias reaches almost –3° versus –2.0° actual. After the dynamic maneuvering, the bias estimation settles close to the actual values. Figures 5.34 and 5.35 demonstrate that parameter estimation and eigenvalue estimation are unaffected by the introduction of the biases. The parameter and eigenvalue convergence is similar to the no bias case shown in Figures 5.8 and 5.9

Snapshots of path prediction with and without rudder and roll bias are displayed in Figure 5.36. The close overlay of the predicted and actual paths indicates proper bias estimation and path prediction.

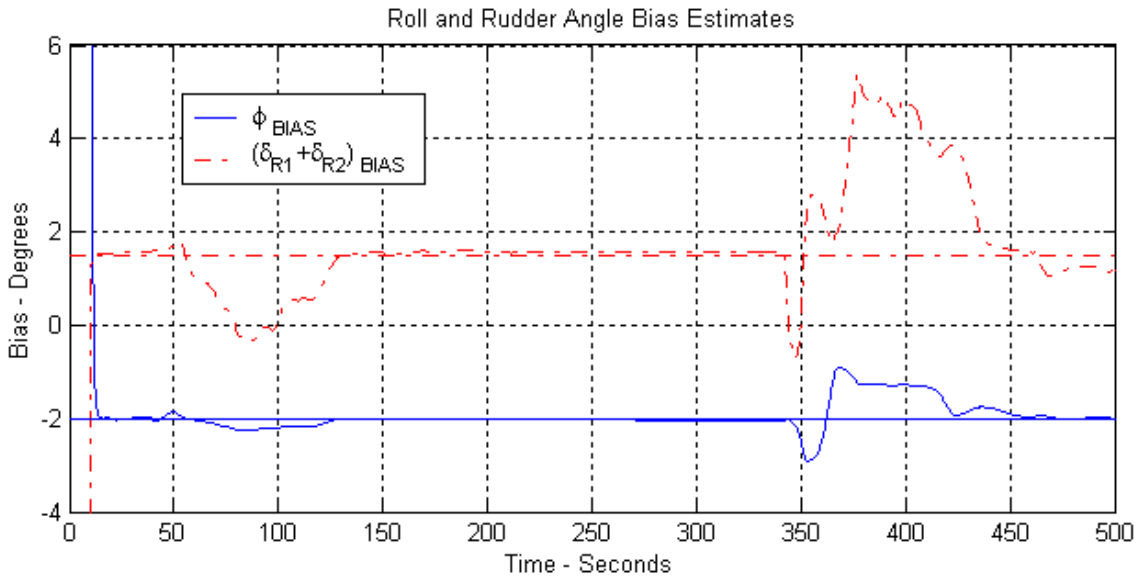


Figure 5.33 Roll and Rudder Angle Bias Estimation ($\lambda = 0.99$)

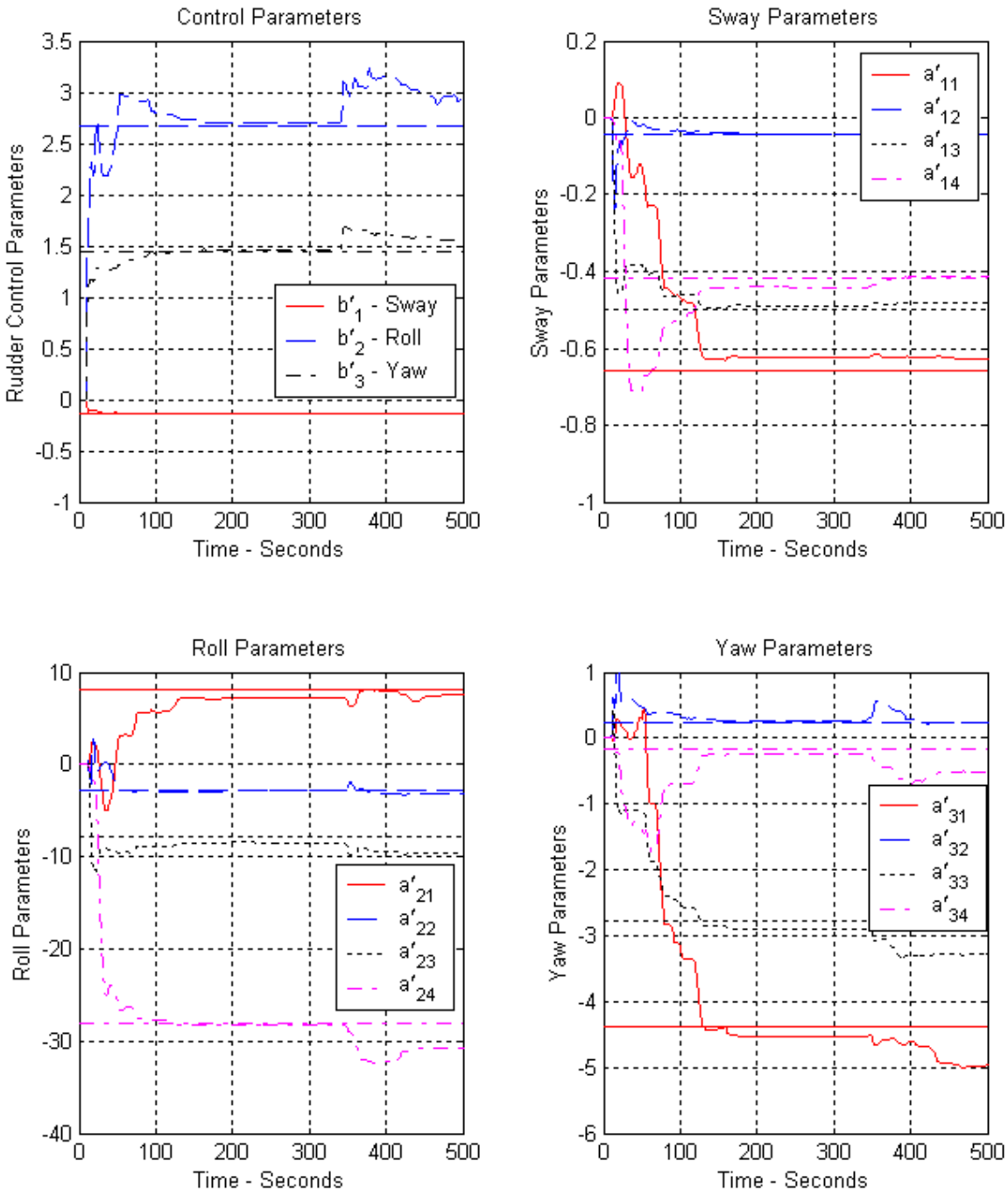


Figure 5.34 Sway–Roll–Yaw Parameter Estimation with Simulated –2° Roll Bias and +1.5° Rudder Bias Compared to Linear Containership Parameters (U_o = 7 m/s) (λ = 0.99)

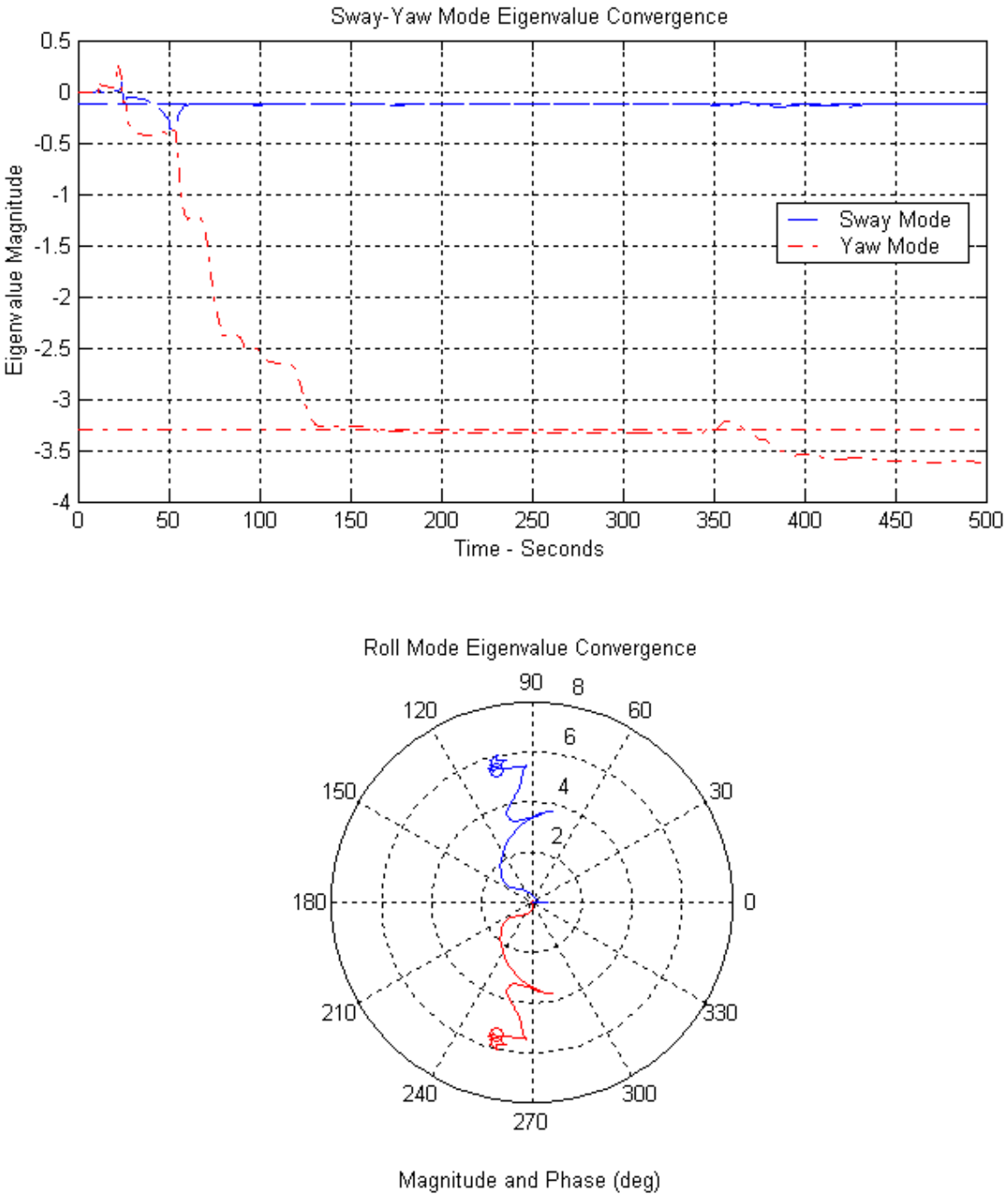
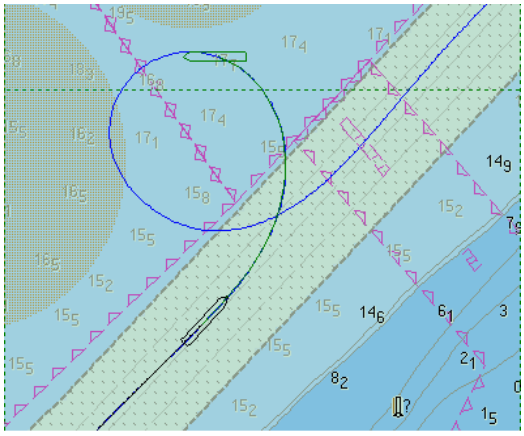
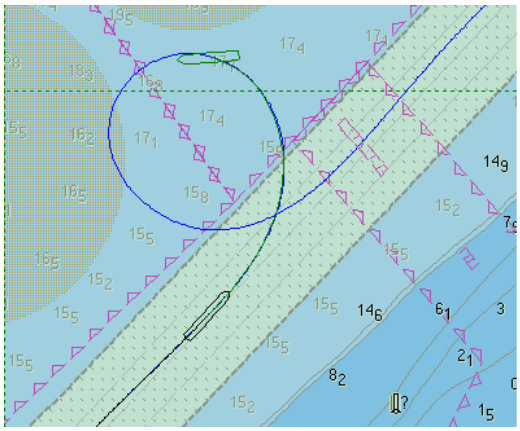


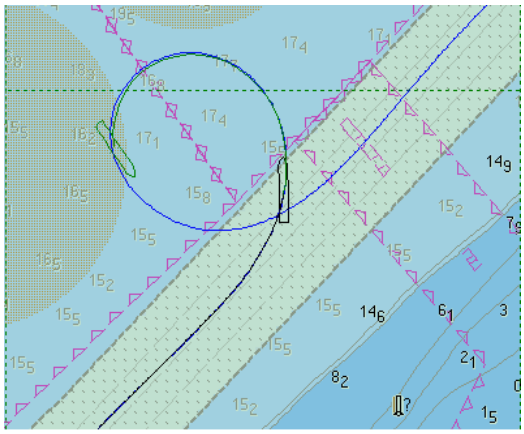
Figure 5.35 Sway–Roll–Yaw Eigenvalue Estimation with Simulated –2° Roll Bias and +1.5° Rudder Bias Compared to Linear Containership Eigenvalues (U_o = 7 m/s)



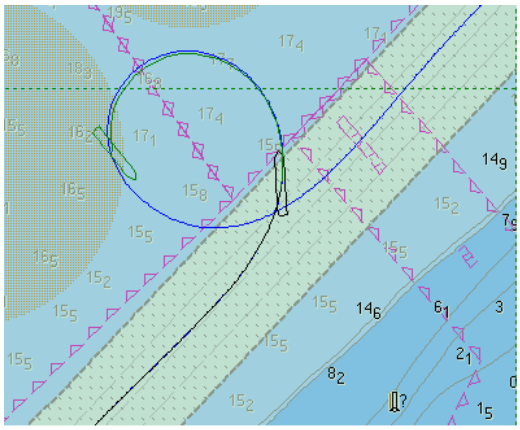
With Roll/Rudder Bias (170 sec)



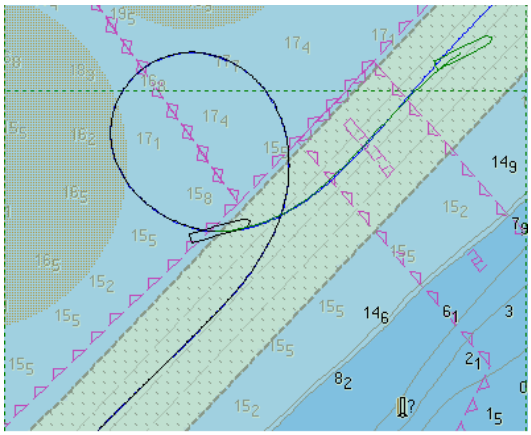
No Bias Truth Model (170 sec)



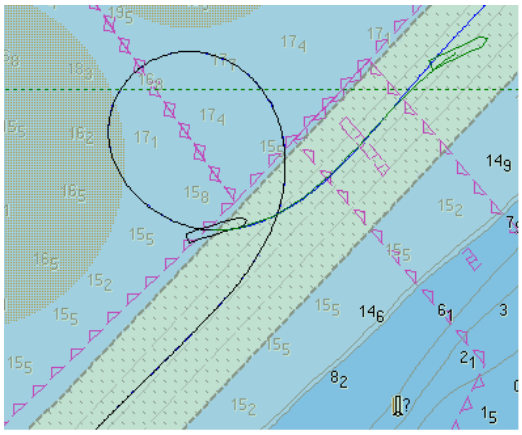
With Roll/Rudder Bias (230 sec)



No Bias Truth Model (230 sec)



With Roll/Rudder Bias (405 sec)



No Bias Truth Model (405 sec)

Figure 5.36 Path Prediction with Bias Estimation Compared to Truth Model without Bias

F. SURGE PARAMETER ESTIMATION VERIFICATION

Up until now, there has been little discussion of the surge estimation algorithm because most of the sway–roll–yaw model suitability investigation was performed using a constant surge velocity truth model ($u[k] = u[k-1]$). Surge parameter estimation requires verification prior to processing sea trials data under real–world conditions.

The non-linear surge equation parameter estimation algorithm was verified using the non-linear containership model. Satisfactory performance is indicated by parameter convergence and stability. Unfortunately, direct comparison with the actual parameters in the containership model is not possible. The non-linear, linear in parameter surge equation is a different model structure than the NL Fossen containership model. However, parameter convergence and subjective observation of path prediction indicates the selected model demonstrates sufficient fidelity for path prediction.

The system was initialized to the settings in Table 5.6. Control inputs and states for driving surge parameter verification are displayed in Figure 5.37. Converged surge equation parameters are shown in Figure 5.38. The parameters settled down by $t = 225$ seconds and remained relatively stable through the long 20–degree left rudder input. At $t = 340$ seconds, simultaneous large RPM transients and rudder reversals caused additional parameter convergence. The parameters then remained stable for the rest of the data run. Subjective observation of the predicted path indicates good performance.

Parameter	***Setting***
Truth Data	Non-Linear Containership Model – Uo = 7 m/s
Truth Model Time Step	0.2 s (RK2 Integration)
RLS/ELS – *v-p-r* Model	ELS with *v*, *p* and *r* Bias Enabled
RLS/ELS Time Step	0.2 s
Exponential Forgetting	0.99 for *v-p-r* Model
Directional Forgetting	On
Covariance Reset	Auto @ cond $P_{vpr} > 0.5 \times 10^6$, $P_{vpr} = P_{vpr} + 10*I$
Surge Model	Non-Linear Surge ($\lambda_u = 0.995$)

Table 5.6 Settings for Investigation of Surge Parameter Estimation

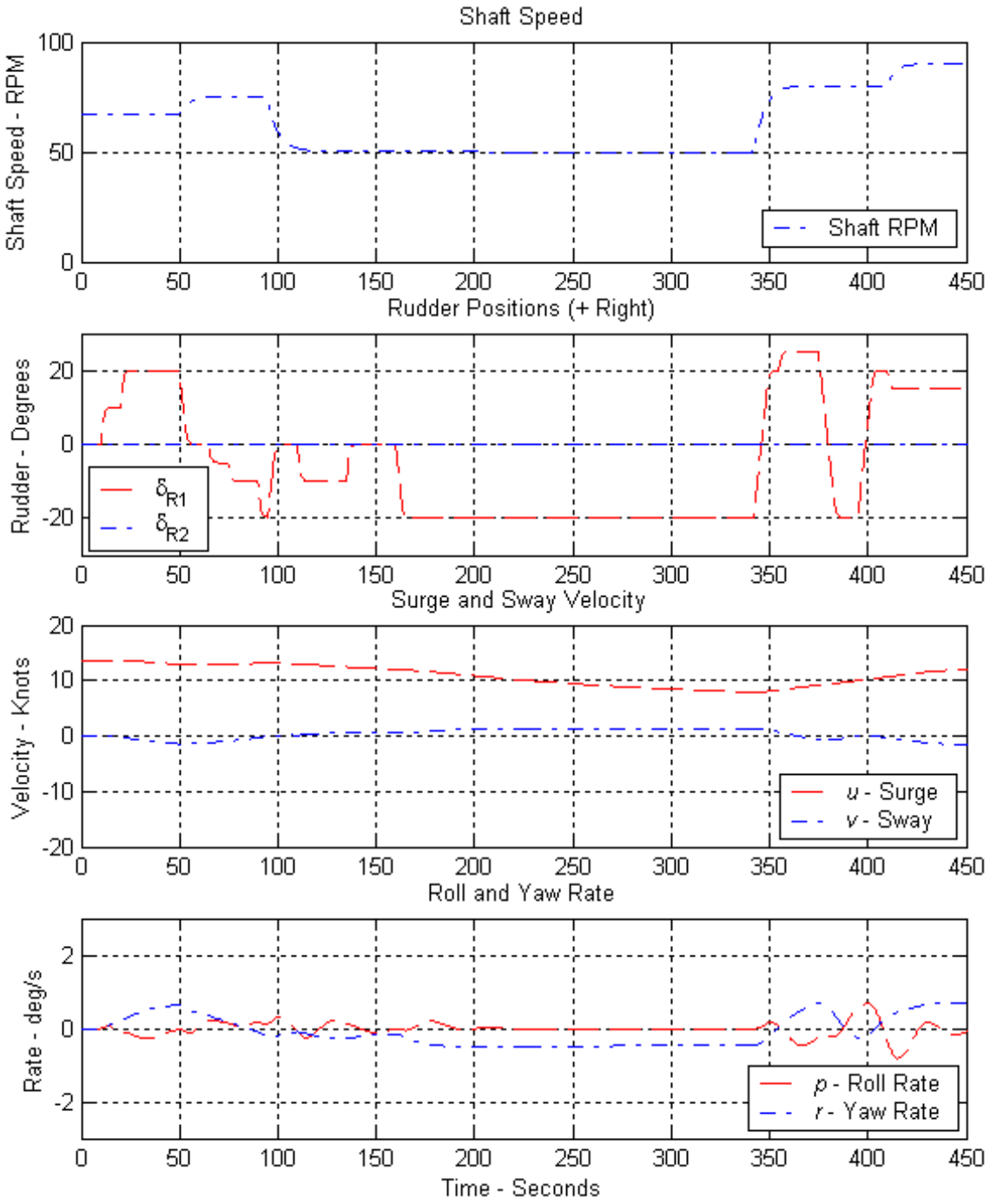


Figure 5.37 Vessel States for Verification of Surge Parameter Estimation

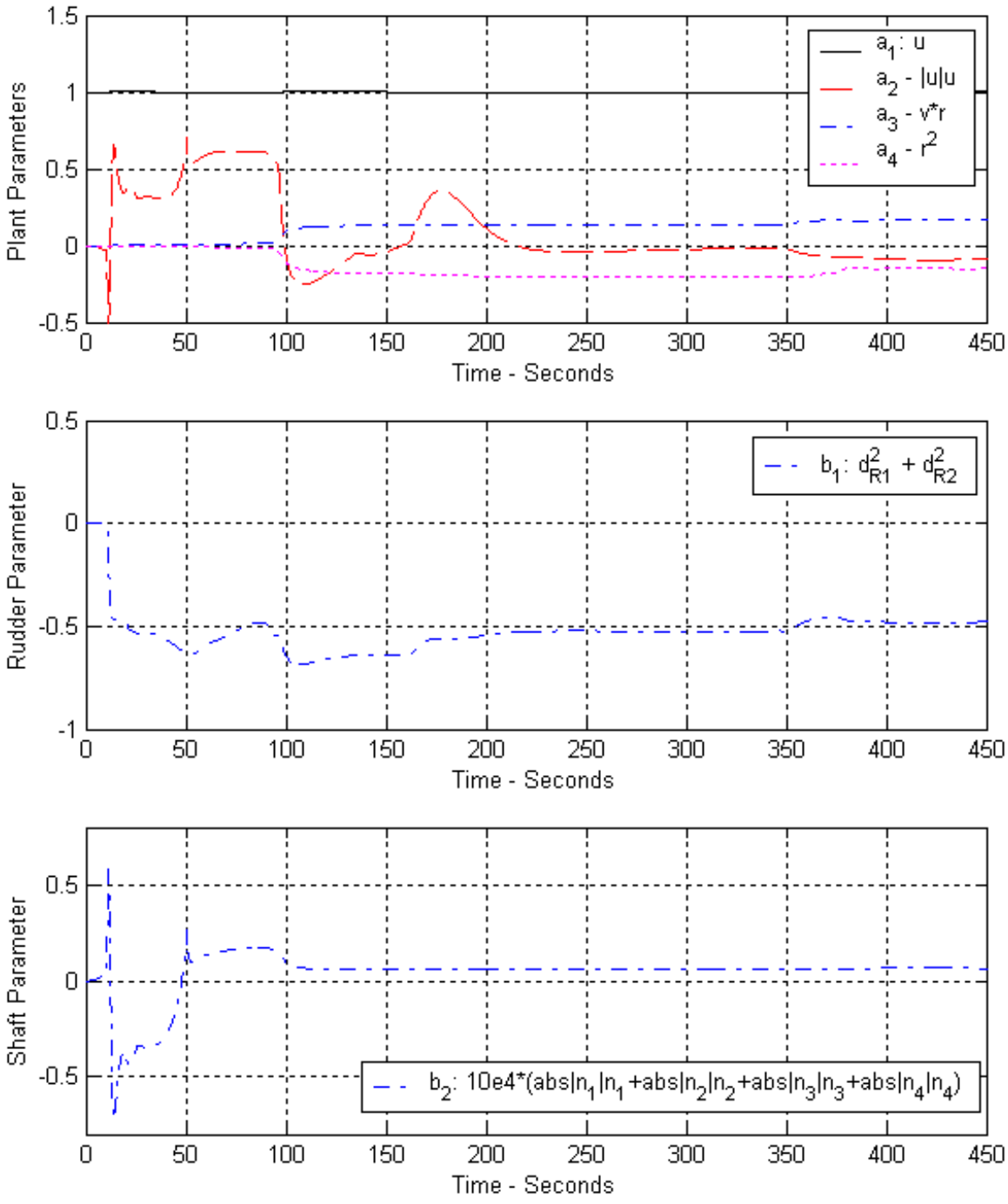


Figure 5.38 Surge Parameter Estimation Driven by Truth Data from Fossen Non-linear Containership Model ($\lambda = 0.99$)

G. PATH PREDICTION PERFORMANCE USING SEA TRIALS DATA

After verification of the basic algorithms, overall system performance was evaluated using real–world data from the USCGC HEALY (WAGB 20) Performance and Sea Trials [Ref. 5]. Parameter estimation and path prediction results are presented for the "4030" data file. The "4030" data file contains approximately 450 seconds of vessel states during a series of 50° zigzags using 20° of rudder at maximum speed (16 knots). This data file was chosen because it provides sufficient excitation to allow parameter convergence in a reasonable time. The system was initialized to the settings in Table 5.7. The control inputs and states are displayed in Figure 5.39. The actual vessel path is displayed in Figure 5.40. It is important to note that the USCGC HEALY sea trials were conducted in the open Gulf of Mexico. However, the data are overlaid on an available Chesapeake Bay approach electronic chart image. The image pixels were remapped to remove latitude distortion. Two sets of results are presented. The first results are based on unfiltered raw data. The second results are based on a low–pass filtered version of the same data file (4–parameter FIR filter: F_c = 0.2 Hz).

Parameter	***Setting***
Truth Data	USCGC HEALY (WAGB 20) – "4030" Data File
Truth Model Time Step	1.0 Hz (Actual Vessel Data)
RLS/ELS – *v*-*p*-*r* Model	ELS with *v*, *p* and *r* Bias Enabled
RLS/ELS Time Step	1 Hz
Exponential Forgetting	0.995 for *v*-*p*-*r* Model
Directional Forgetting	On
Covariance Reset	Auto @ cond $P_{vpr} > 0.5 \times 10^6$, $P_{vpr} = P_{vpr} + 10*I$
Surge Model	Non-Linear Surge (λ_u = 0.995)

Table 5.7 Settings for USCGC HEALY Sea Trials Evaluation

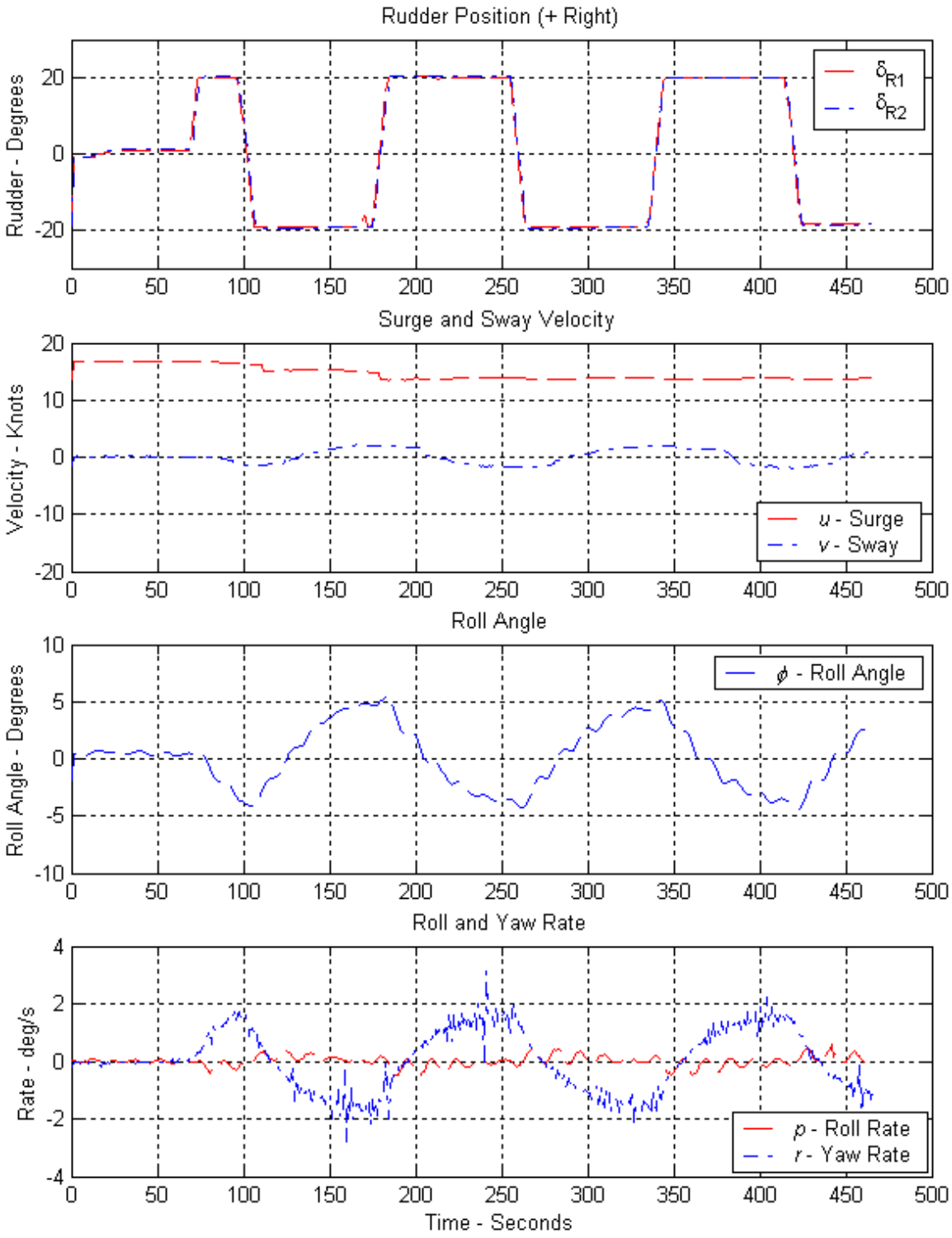


Figure 5.39 USCGC HEALY "4030" Maneuver Unfiltered States

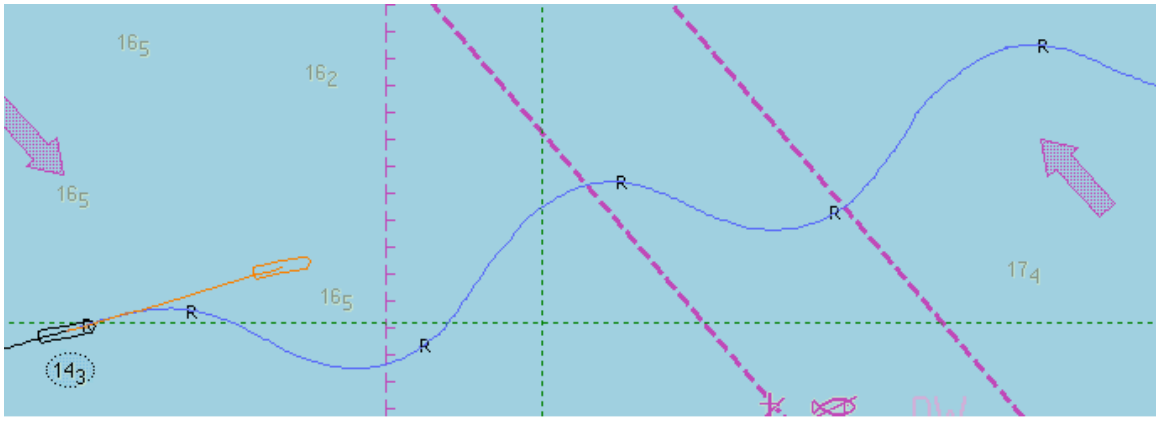


Figure 5.40 USCGC HEALY "4030" Maneuver Overlaid on Navigational Chart

1. Parameter Estimation and Path Prediction Using Unfiltered "4030" Data File

This first set of results is based on allowing the sway–roll–yaw ELS algorithm to process noisy, unfiltered 1–Hz data from the USCGC HEALY "4030" file. The data set includes noise generated by differentiating angle states into angle rates.

a. Sway-Roll-Yaw Parameter Estimation and Path Prediction Using Unfiltered "4030" Data File

ELS sway–-roll–yaw parameter estimation is displayed in Figure 5.41 for the first pass of the unfiltered "4030" data file. Estimation progress is considerably slower than previous examples using math model truth data. The primary difference in convergence rates is probably due to the slow 1–Hz sea trials data rate versus the 5–Hz math model rate. In addition, the sea trials data is corrupted by real–world noise, including measurement and environmental noise.

A second pass of the "4030" data file was made using the parameters and covariance from the end of the first pass as the initialization point. The primary objective of the second pass was to investigate parameter and eigenvalue stability. Parameter estimation during the second pass is displayed in Figure 5.42. The parameters remained reasonably steady following initial convergence from the first pass, indicating a good match with the data.

Eigenvalue convergence is depicted in Figures 5.43 and 5.44. The eigenvalues begin to stabilize towards the end of the first data run. However, the complex (oscillatory) roll mode eigenvalues never really form. It appears that the parameter estimation algorithm started tracking the high frequency noise in the unfiltered roll and yaw rate terms instead of the oscillatory rolling motion. This is evidenced by the very high a'_{34} yaw rate due to roll angle term. The roll mode eigenvalues continue to be split and real in the second pass.

The roll and rudder bias terms are displayed in Figures 5.45 and 5.46. Bias values are reasonably steady after initial parameter convergence. Steady biases indicate a reasonable fit of the linear *v–p–r* model to the non-linear sea trials data. The fairly steady –2.5° rudder bias is a combination of sensor bias and the tendency of the vessel to weathercock into the wind (020° true at 5 knots).

Covariance matrix trace and condition data are displayed in Figures 5.47 and 5.48. The sway–roll–yaw covariance matrices are well behaved with no significant creep in the trace or condition numbers. Noise excitation appears beneficial for limiting covariance growth. The trace and condition numbers for each degree of freedom are better behaved than in the previous runs using noise free math model truth data.

Path estimation snap shots are displayed in Figure 5.49. The first and second passes are compared side by side at identical times. The first snapshot at t = 267 seconds depicts path prediction just as the rudder reaches left 20°. The second snapshot at t = 282 seconds depicts path prediction progress 15 seconds later. Snapshots for both data passes indicate nice path prediction. Very little qualitative difference is observed between the first and second passes. This indicates that suitable path prediction may be achieved prior to parameter convergence.

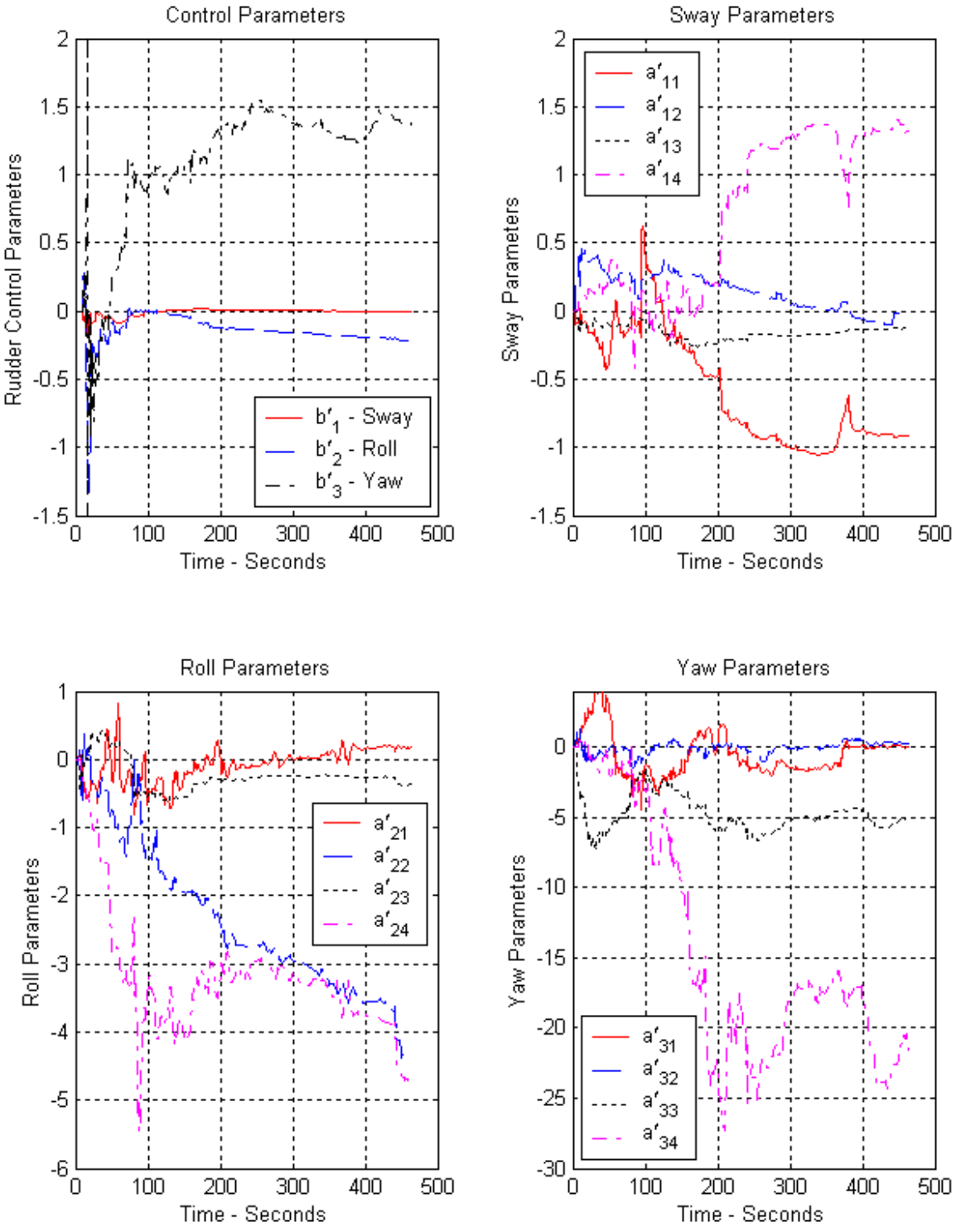


Figure 5.41 First Pass Sway–Roll–Yaw Parameter Estimation using Unfiltered USCGC HEALY "4030" Data File (U_o = 8 m/s) (λ = 0.995)

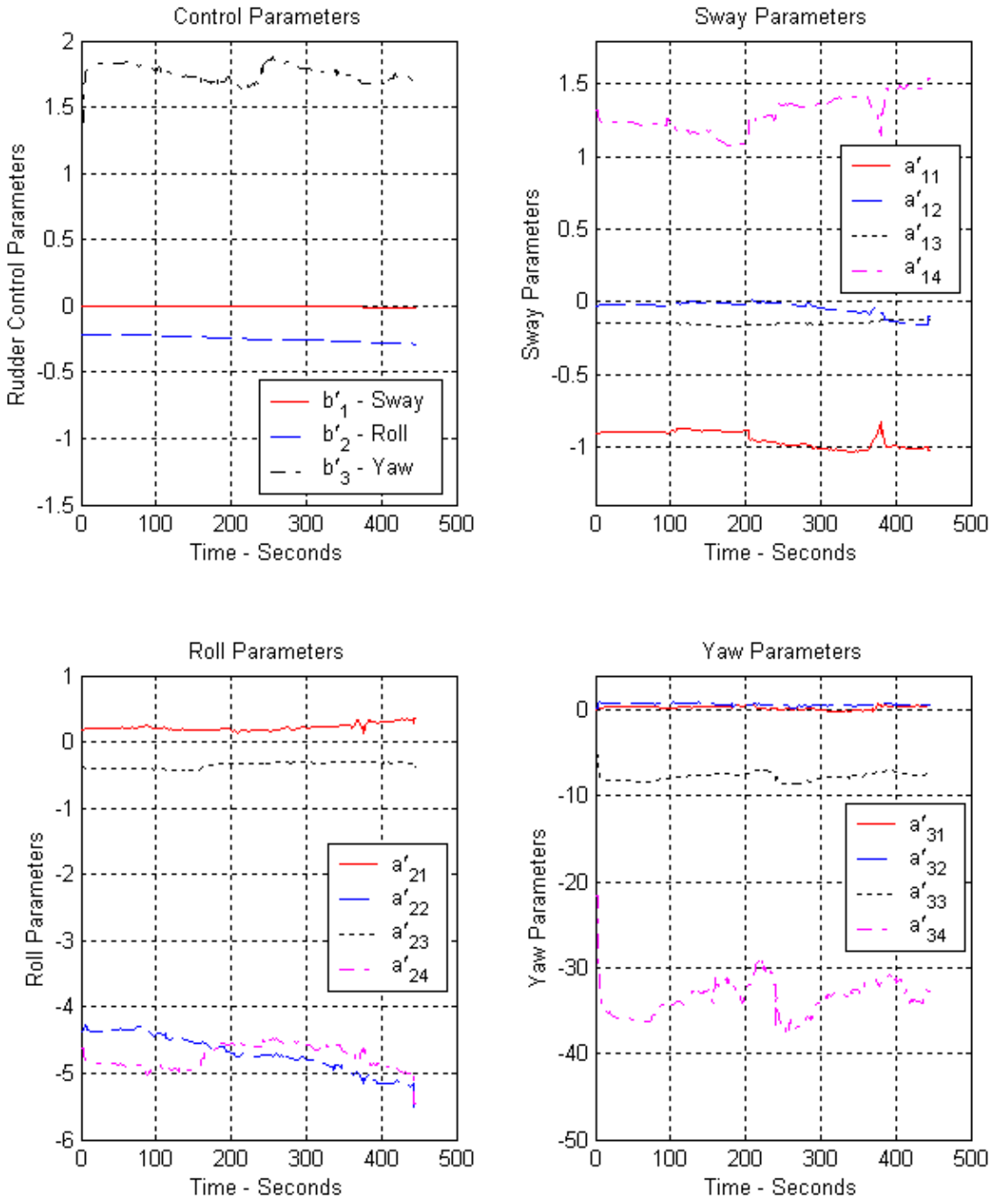


Figure 5.42 Second Pass Sway–Roll–Yaw Parameter Estimation using Unfiltered USCGC HEALY "4030" Data File (U_o = 8 m/s) (λ = 0.995)

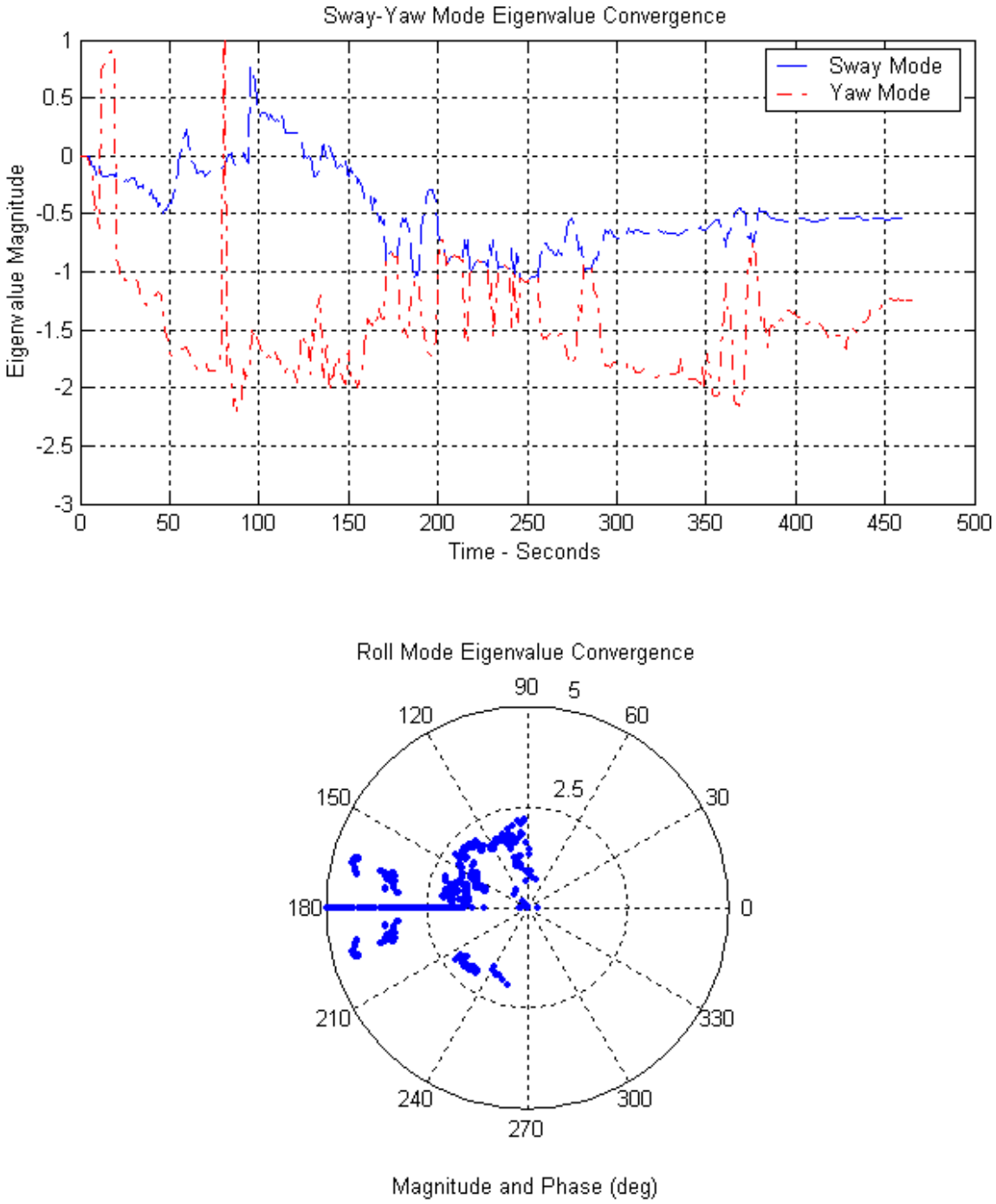


Figure 5.43 First Pass Sway–Roll–Yaw Eigenvalue Estimation using Unfiltered USCGC HEALY "4030" Data File (U_o = 8 m/s)

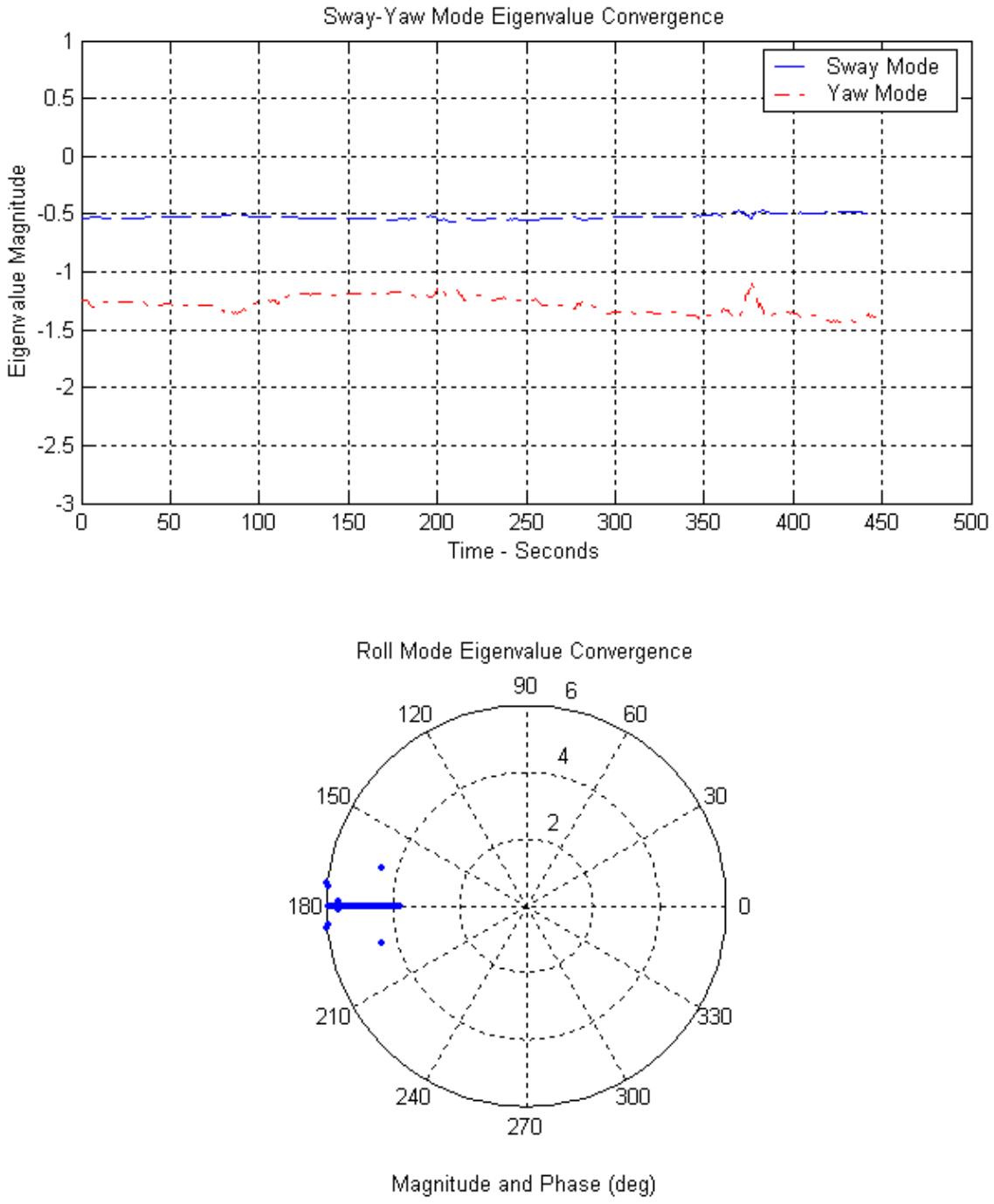


Figure 5.44 Second Pass Sway–Roll–Yaw Eigenvalue Estimation using Unfiltered USCGC HEALY "4030" Data File (U_o = 8 m/s)

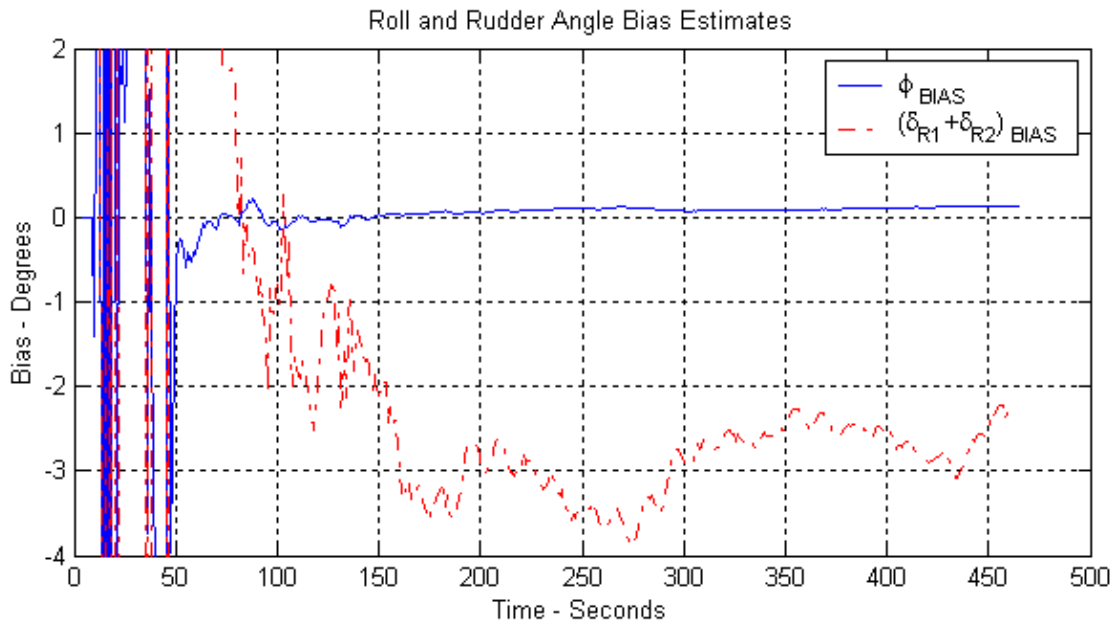


Figure 5.45 First Pass Rudder and Roll Angle Bias Estimation using Unfiltered USCGC HEALY "4030" Data File (U_o = 8 m/s)

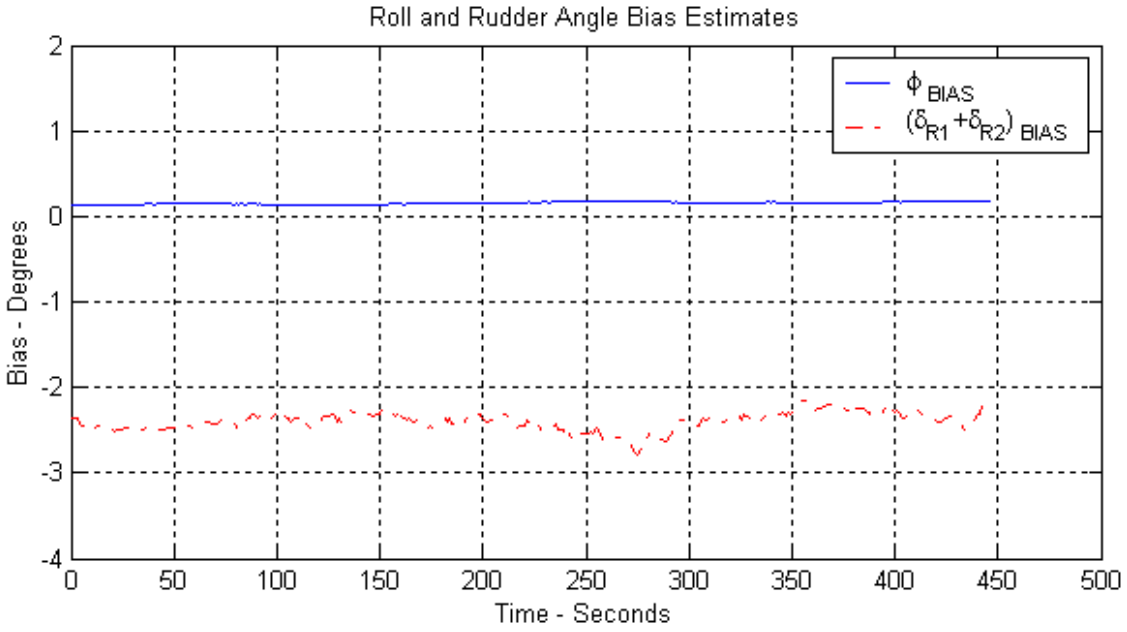


Figure 5.46 Second Pass Rudder and Roll Angle Bias Estimation using Unfiltered USCGC HEALY "4030" Data File (U_o = 8 m/s)

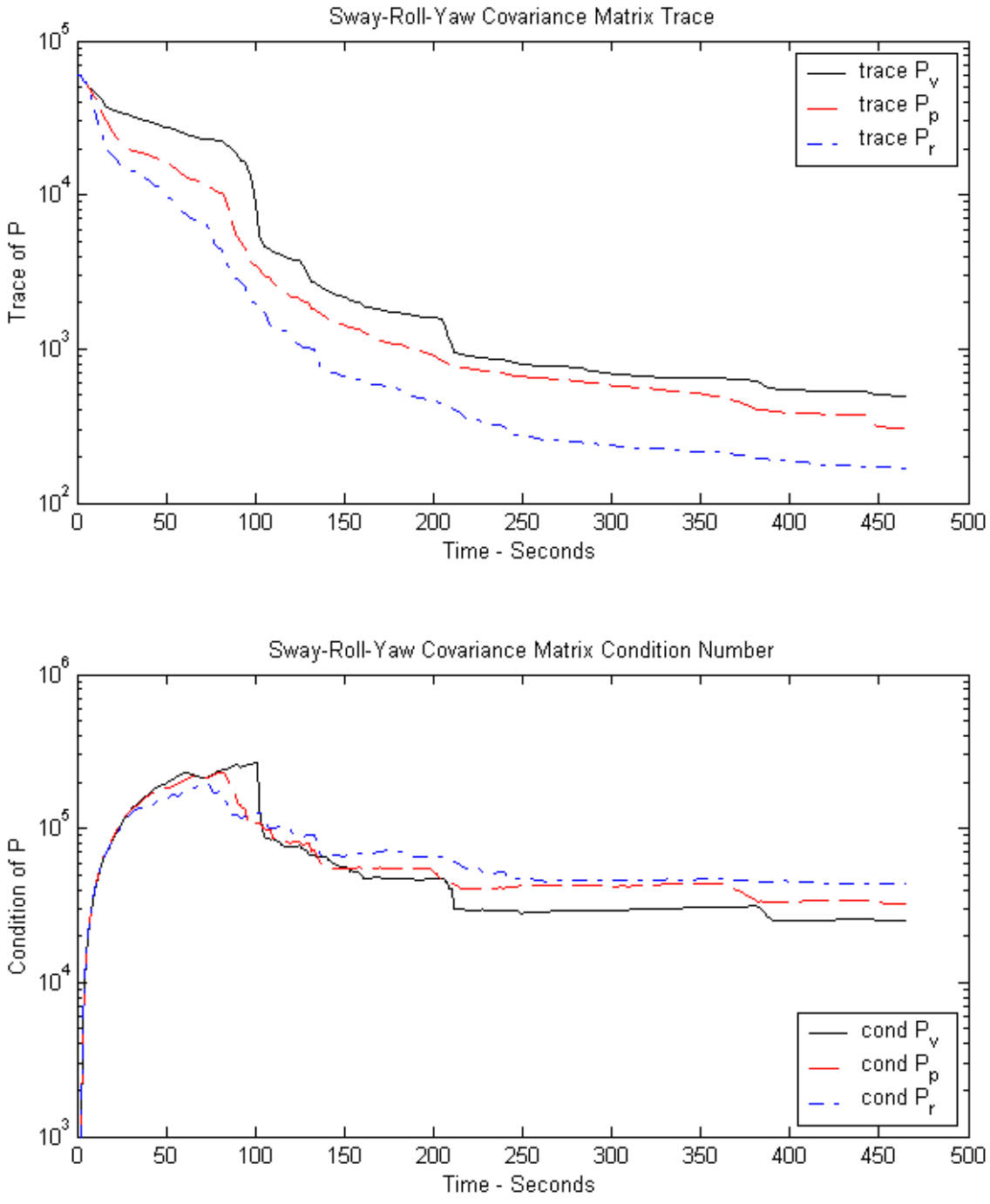


Figure 5.47 Trace and Condition Number of Covariance Matrices Following First Pass of Unfiltered "4030" Data File

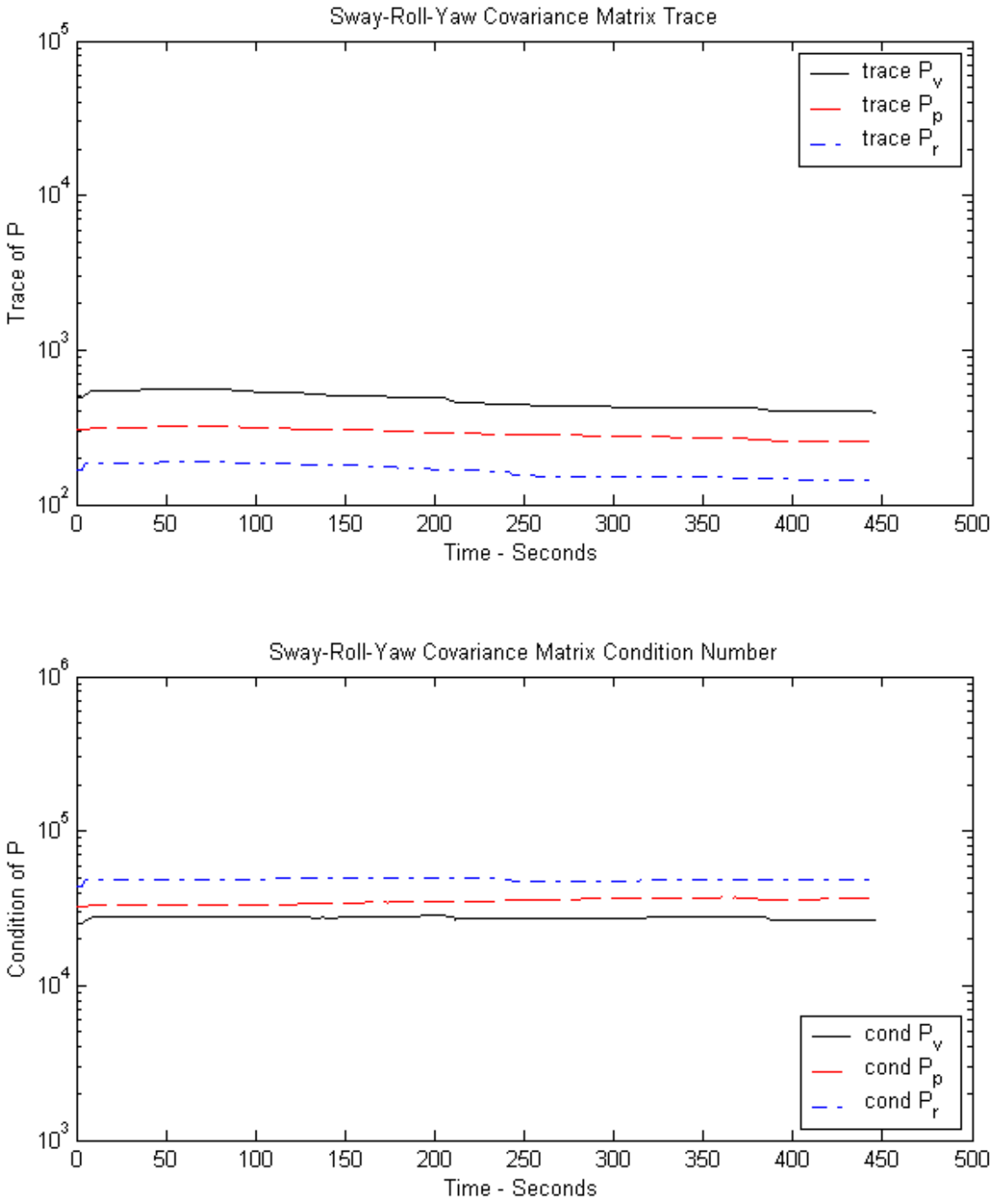
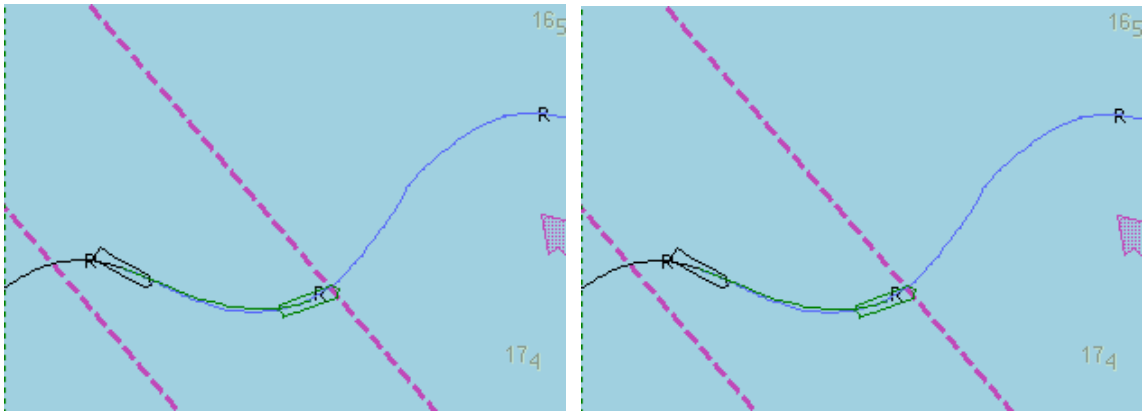
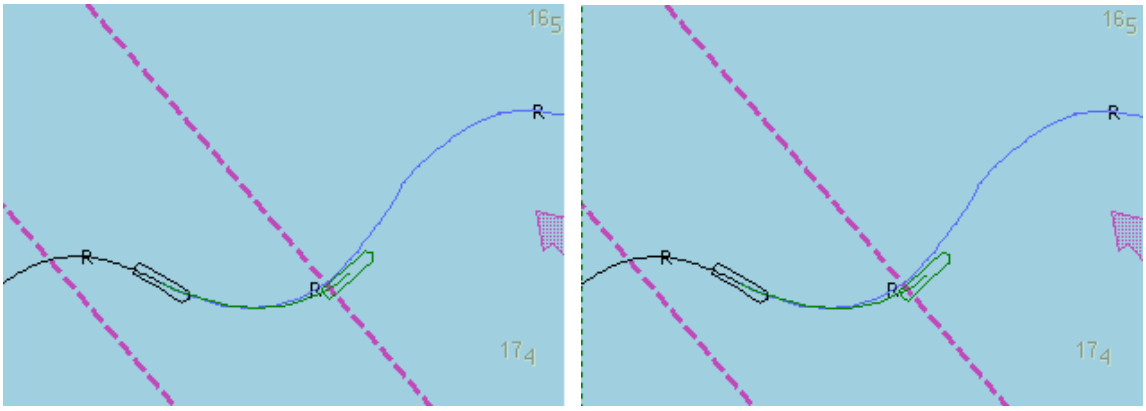


Figure 5.48 Trace and Condition Number of Covariance Matrices Following Second Pass of Unfiltered "4030" Data File



1st Pass (t = 267 sec) 2nd Pass (t = 282 sec)



1st Pass (t = 267 sec) 2nd Pass (t = 282 sec)

Figure 5.49 USCGC HEALY Zig-Zag Path Prediction for First and Second Pass of Unfiltered "4030" Data File

b. Surge Parameter Estimation Using Unfiltered "4030" Data File

Non-linear surge equation parameter identification was also evaluated while processing the unfiltered "4030" data file. The states involved are displayed in Figure 5.50. Notice the independent nature of shaft RPM for the twin–screw vessel.

Surge parameter estimation for pass one and two of the "4030" data file is displayed in Figures 5.51 and 5.52 respectively. The parameters do not remain as steady as for the sway–roll–yaw models. The nice path prediction in Figure 5.49 indicates good performance despite drifting surge equation parameters.

Drifting surge parameters could be due to a number of reasons. First, there was very little excitation of the shaft RPM. Other than some bogging down during the turns, the shaft RPM stayed close to the fixed 150 RPM command. Second, there was no directional forgetting implemented in the surge RLS model. Covariance and gain increased during the initial non-maneuvering portion of the run, causing erratic parameter estimation during the start of maneuvering. Directional forgetting should be added the surge equation algorithm. Another problem may be that the surge algorithm only implements RLS. ELS should be added to better control real–world noise.

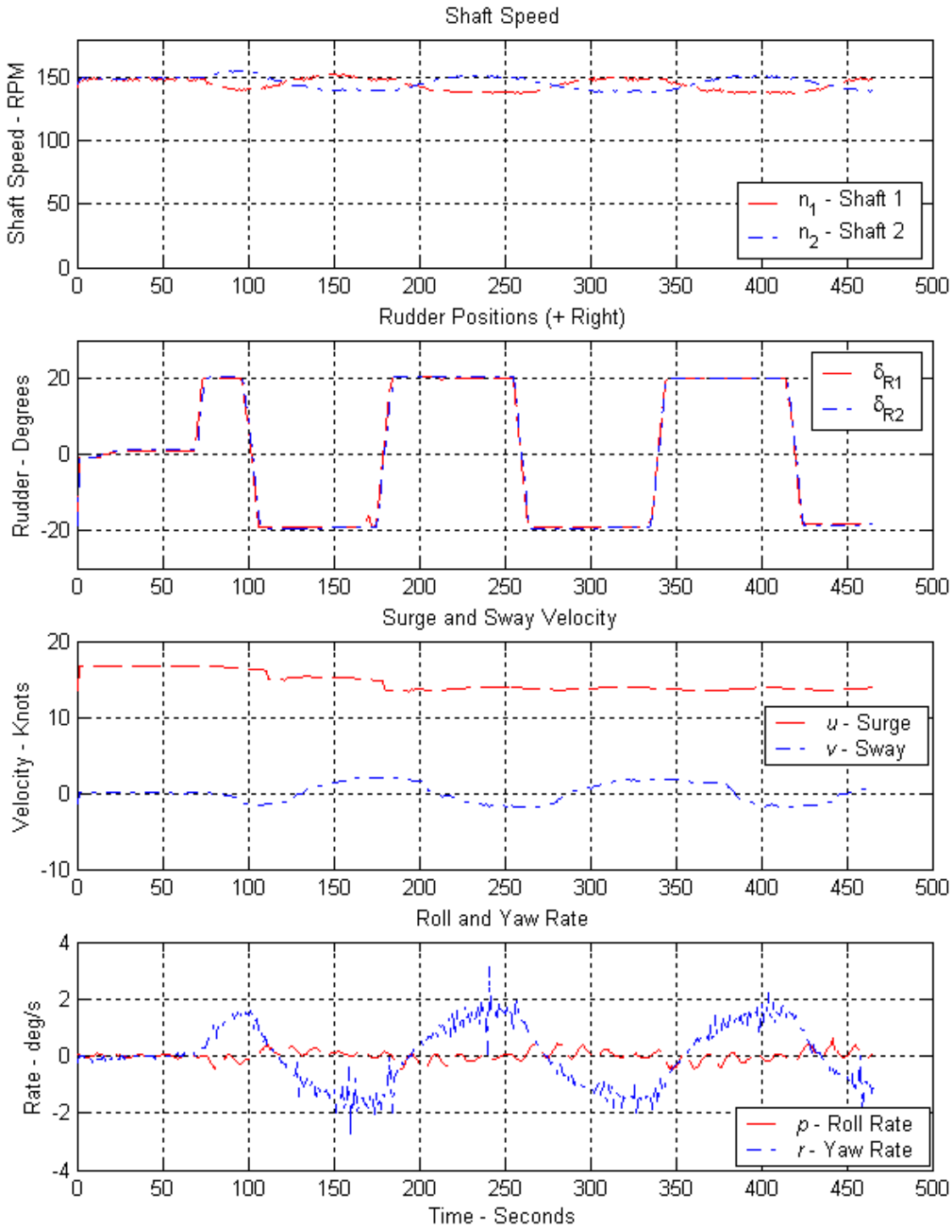


Figure 5.50 Unfiltered "4030" Vessel States for Surge Parameter Estimation

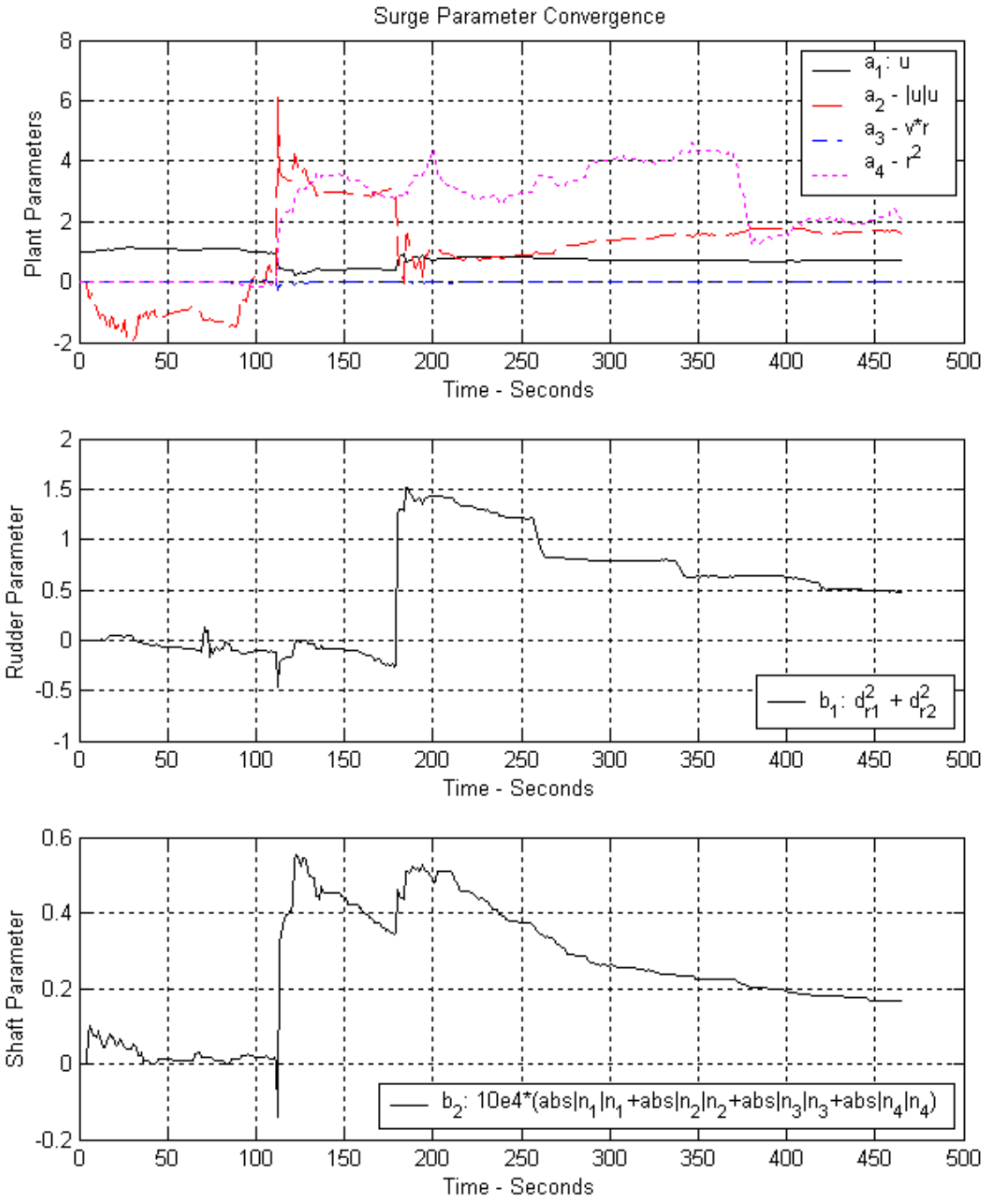


Figure 5.51 Surge Parameter Estimation Following First Pass of Unfiltered "4030" Data File ($\lambda = 0.995$)

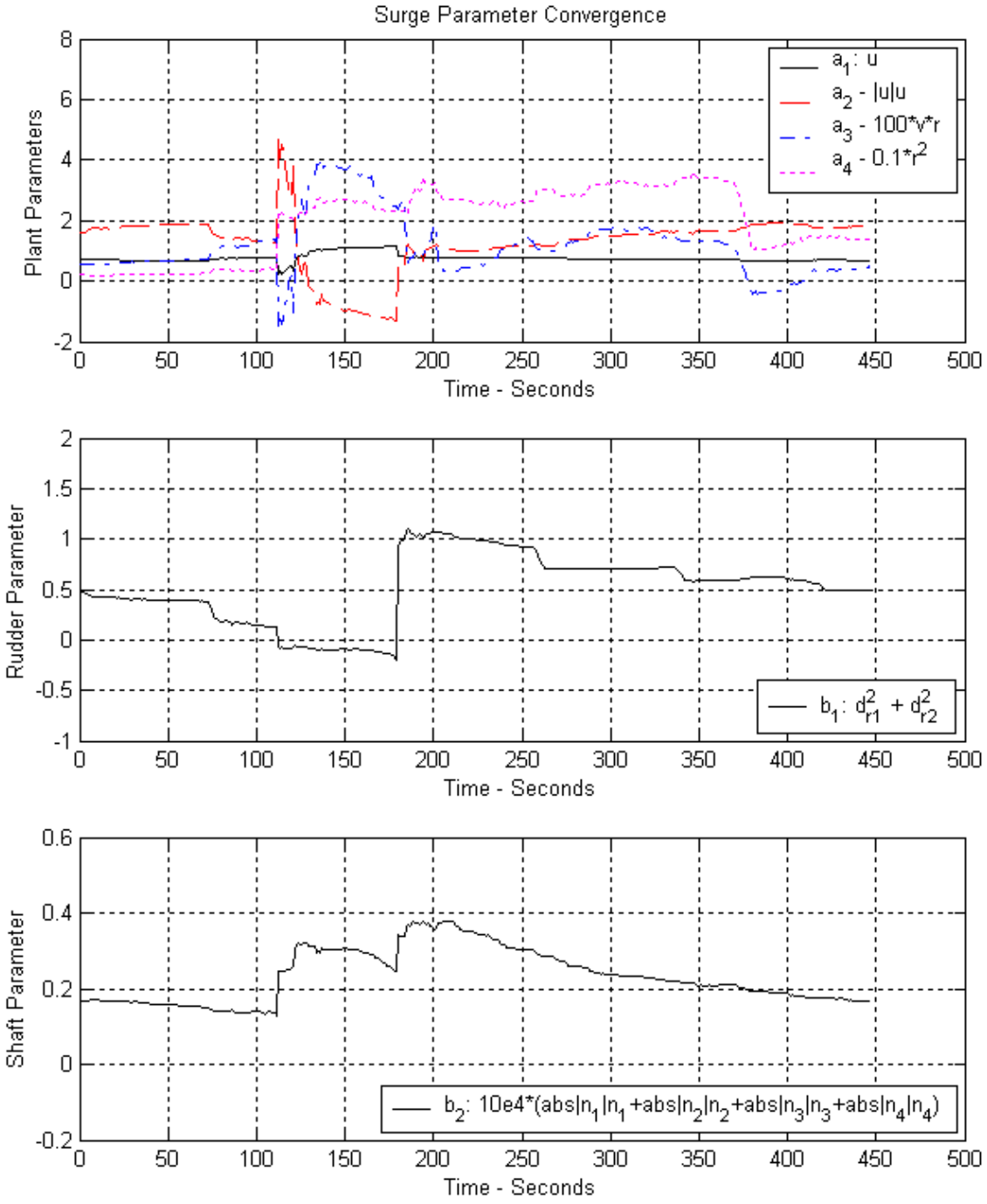


Figure 5.52 Surge Parameter Estimation Following Second Pass of Unfiltered "4030" Data File ($\lambda = 0.995$)

2. Parameter Estimation and Path Prediction Using Low–Pass Filtered "4030" Data File

This second set of results is based on low–pass filtering data from the USCGC HEALY "4030" file in an effort to determine if preprocessing the data improves the parameter estimation and path prediction process. The data set was pre-filtered by a 4–term Hamming FIR filter with a 0.2–Hz cutoff frequency. A low–order filter was selected to provide some data smoothing without introducing unacceptable time delay. The filtered states are displayed in Figure 5.53. Note the large reduction in yaw rate noise compared to the unfiltered yaw rate in Figure 5.39

a. Sway–Roll–Yaw Parameter Estimation and Path Prediction Using Low–Pass Filtered "4030" Data File

ELS sway–roll–yaw parameter estimation for the first and second pass of the low–pass filtered "4030" data file is displayed in Figures 5.54 and 5.55, respectively. Parameter estimation convergence rate and stability are similar that of the unfiltered data file. However, it is clear that the yaw rate terms are no longer tracking noise. For the unfiltered data file, a'_{34} was on the order of 30 to 35. After low–pass filtering the data, the a'_{34} term dropped to 4 to 7.

Eigenvalue convergence is depicted in Figures 5.56 and 5.57. The sway–yaw eigenvalues show much improved stability to that of the unfiltered data runs. After the initial 100 seconds, the sway–yaw modes break apart and stay unique. Additionally, the oscillatory roll mode is well established. Previously, the roll poles split apart and tracked the noise. Now the roll mode stabilizes out. Figure 5.57 clearly shows well-developed stationary roll mode poles.

The roll and rudder bias terms are displayed in Figures 5.58 and 5.59. Bias values are reasonably steady after initial parameter convergence indicating a reasonable fit to the linear v–p–r model. Interestingly, the FIR filtering reduced the average rudder bias from –2.5° to –1.5°.

Covariance matrix trace and condition data are displayed for the first and second passes in Figures 5.60 and 5.61, respectively. The sway–roll–yaw covariance matrices are well behaved with no significant creep in the trace or condition numbers. Co-

variance collapse is slower for the low–pass filtered data file. This confirms that noise is useful for keeping covariance bounded.

Path estimation snap shots are displayed in Figure 5.62. The first and second passes are compared side by side at identical times. The first snapshot at t = 267 seconds depicts path prediction just as the rudder reaches left 20°. The second snapshot at t = 282 seconds depicts path prediction progress 15 seconds later. Both runs are very similar. This indicates that the parameters that were still settling during the first pass had very little impact on path prediction. Also, both runs over–predict the turn rate at t = 267 seconds. In fact, the prediction quality using the filtered data is worse at t = 267 seconds than for the unfiltered data. However, in the next fifteen seconds, the filtered data path prediction settles down and provides superior results. Additional investigation is required to determine why the filtered data provides a poorer prediction immediately after rudder input. Perhaps there is time delay in the system or the 0.2–Hz cutoff frequency was inappropriate.

In general, processing low–pass filtered data removed the tendency of the parameter estimation algorithm to track noise. However, the initial path prediction accuracy suffered compared to unfiltered data. The filtered data provided superior path prediction by 15 seconds into the turn. Alternative filter settings should be investigated to improve the initial response.

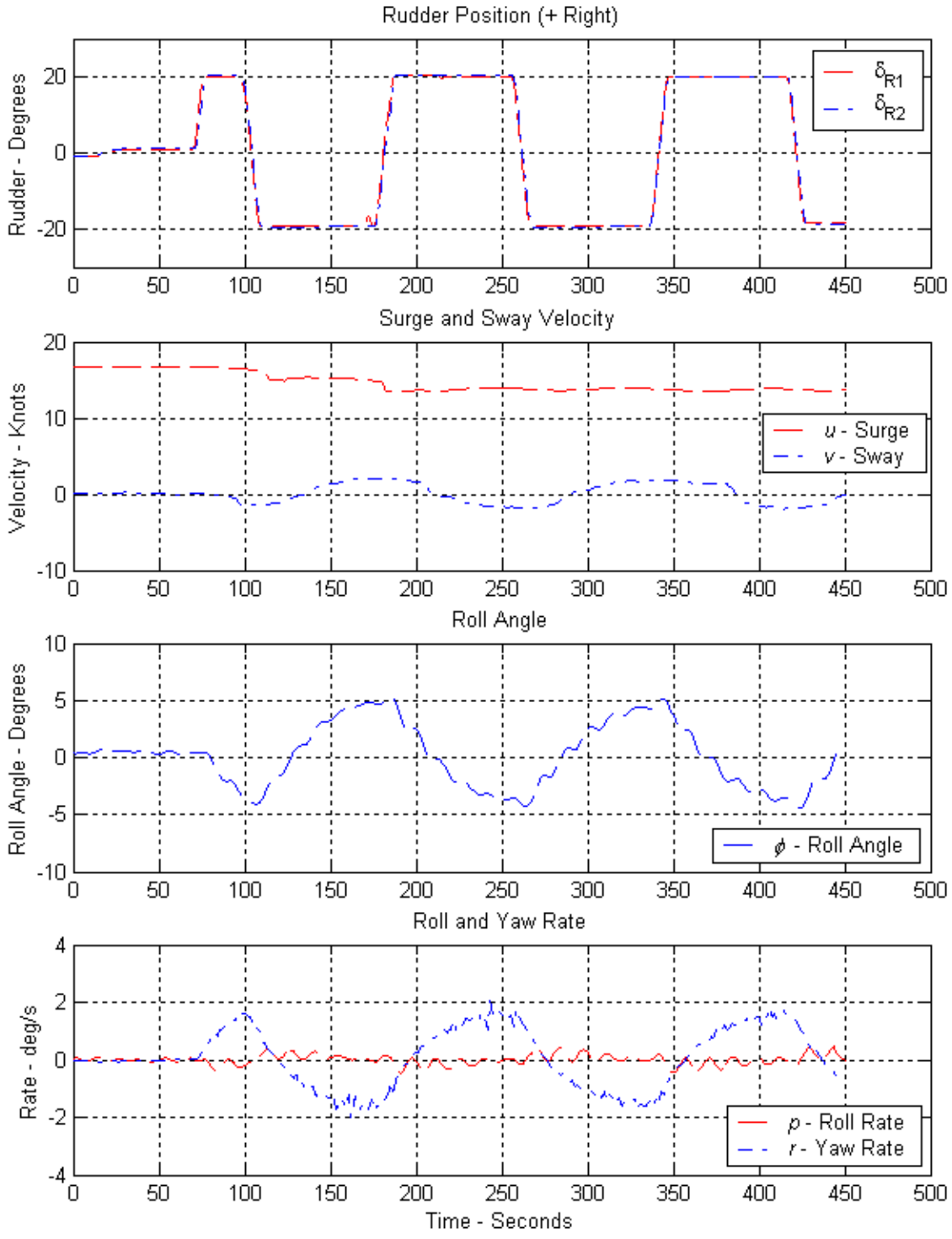


Figure 5.53 USCGC HEALY "4030" Maneuver States Following Low–Pass Filtering

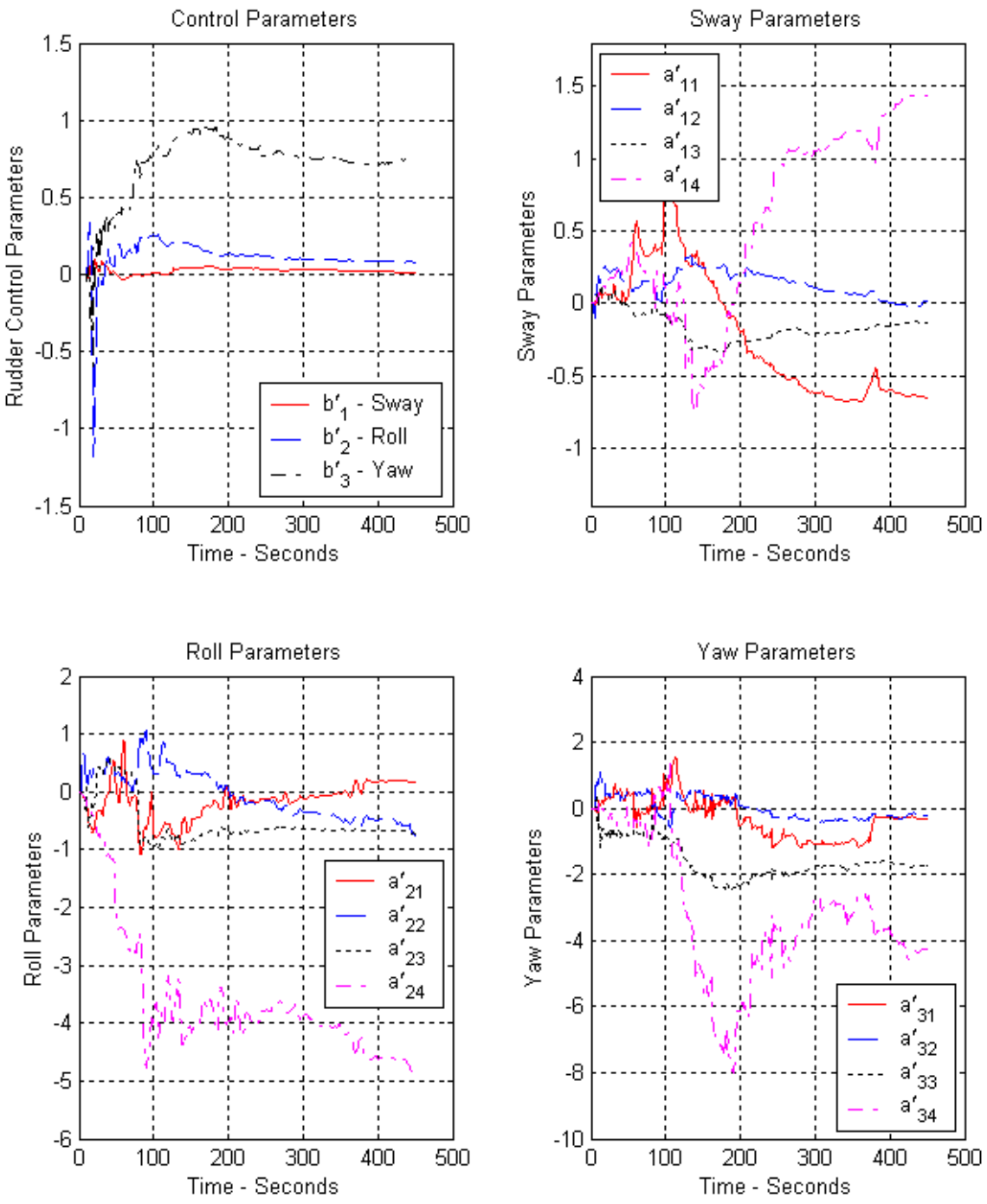


Figure 5.54 First Pass Sway–Roll–Yaw Parameter Estimation using Low–Pass Filtered USCGC HEALY "4030" Data File (U_o = 8 m/s) (λ = 0.995)

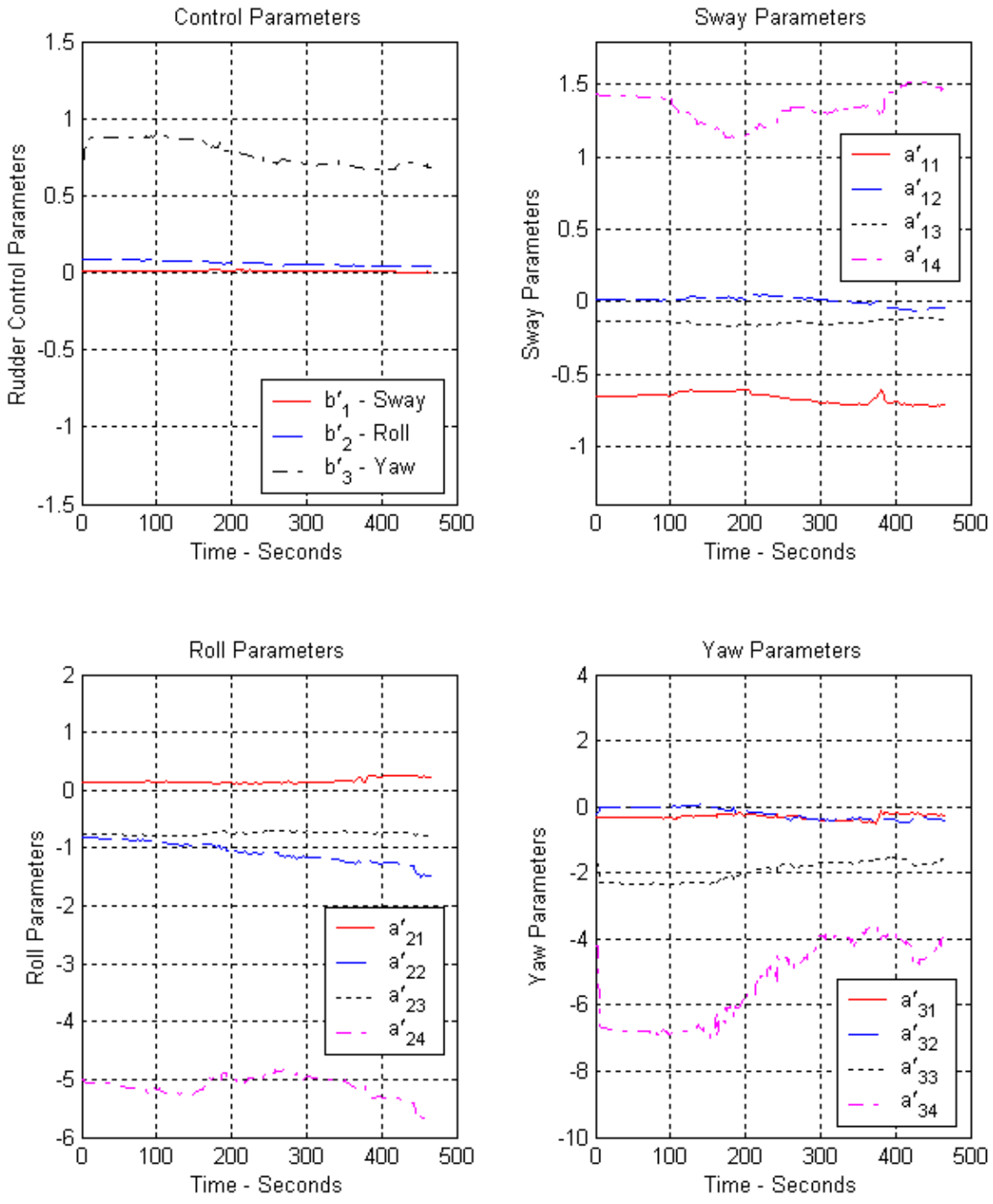


Figure 5.55 Second Pass Sway–Roll–Yaw Parameter Estimation using Low–Pass Filtered USCGC HEALY "4030" Data File (U_o = 8 m/s) (λ = 0.995)

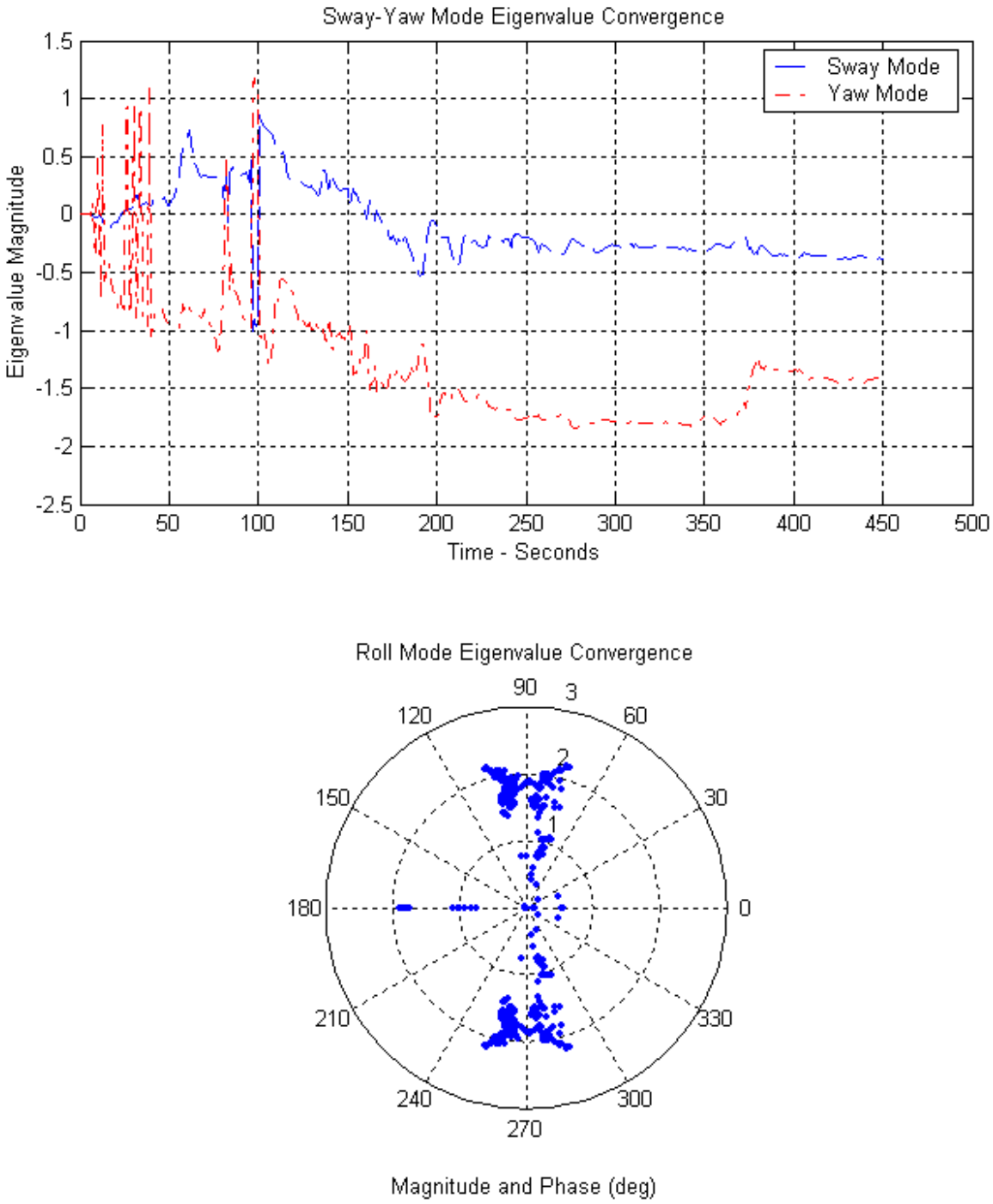


Figure 5.56 First Pass Sway–Roll–Yaw Eigenvalue Estimation using Low–Pass Filtered USCGC HEALY "4030" Data File (U_o = 8 m/s)

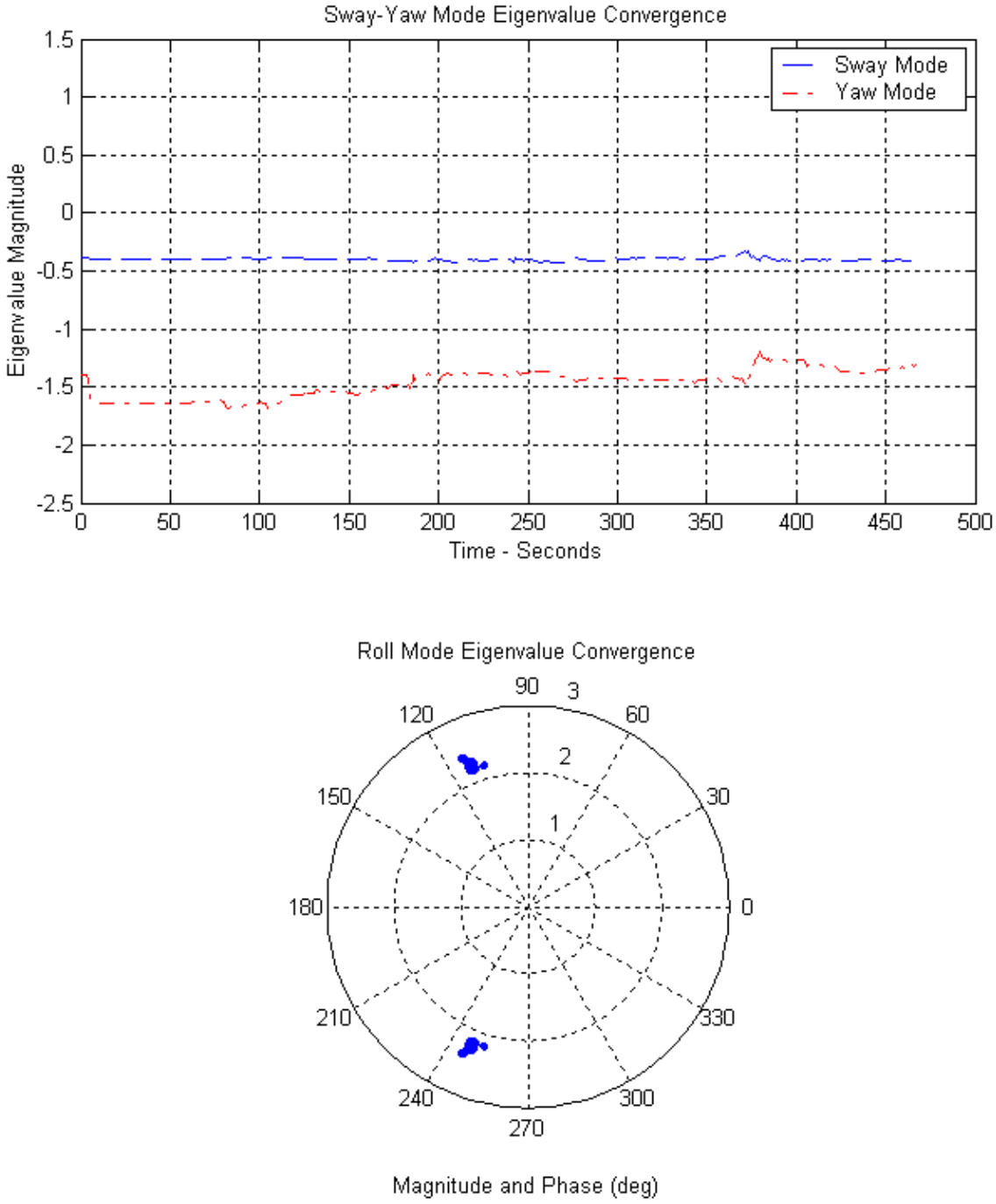


Figure 5.57 Second Pass Sway–Roll–Yaw Eigenvalue Estimation using Low–Pass Filtered USCGC HEALY "4030" Data File (U_o = 8 m/s)

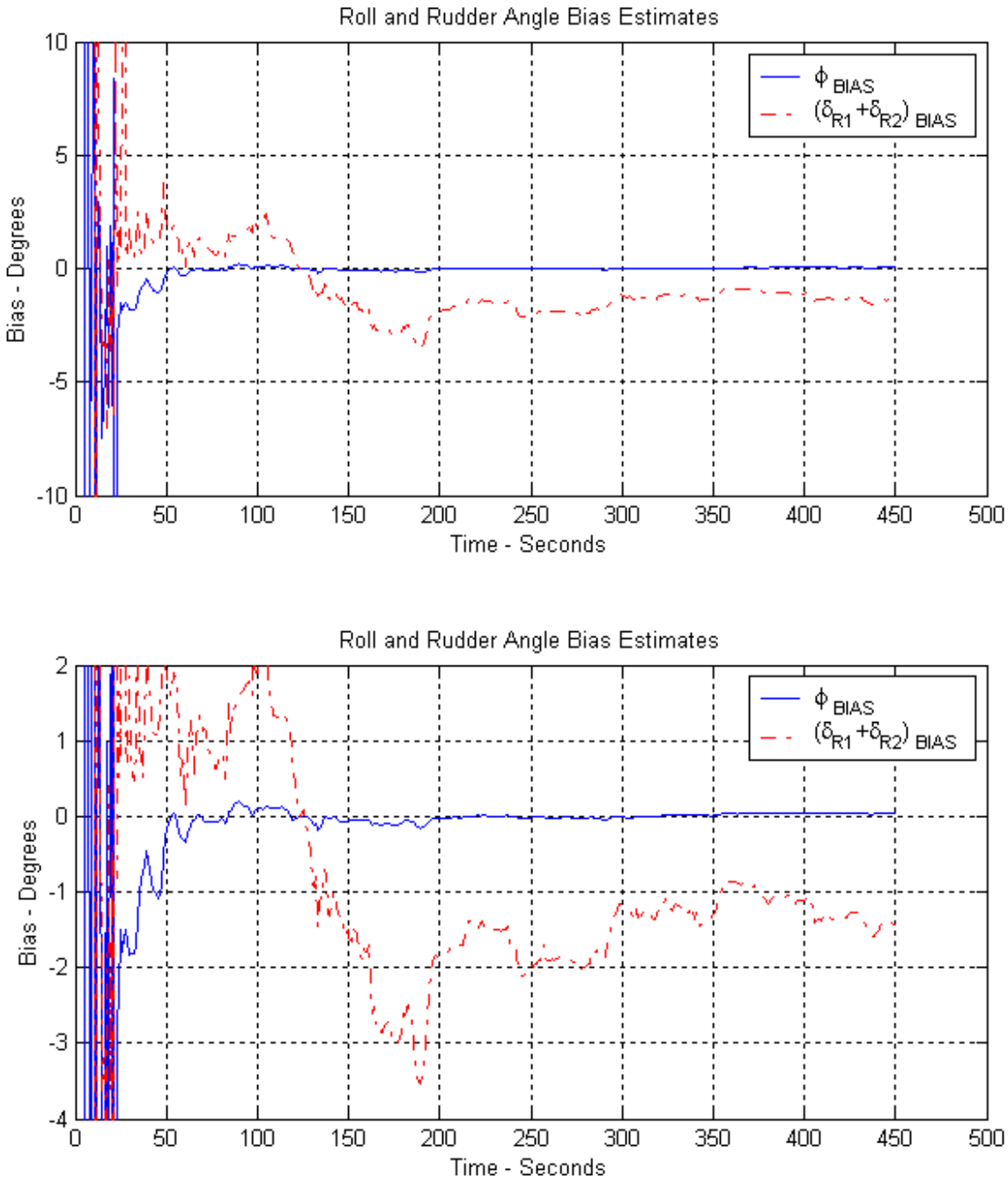


Figure 5.58 First Pass Rudder and Roll Angle Bias Estimation using Low–Pass Filtered USCGC HEALY "4030" Data File (U_o = 8 m/s)

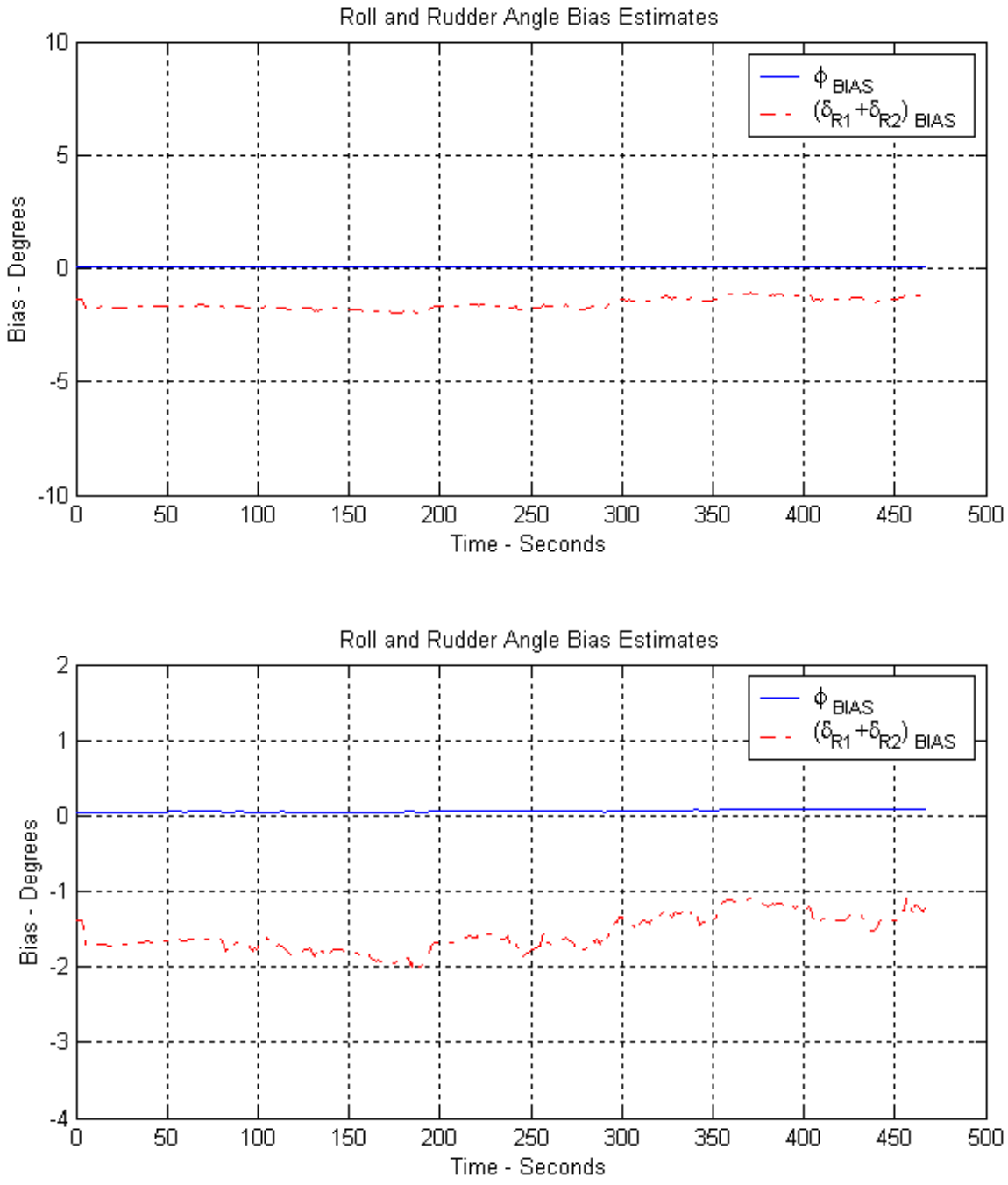


Figure 5.59 Second Pass Rudder and Roll Angle Bias Estimation using Low–Pass Filtered USCGC HEALY "4030" Data File (U_o = 8 m/s)

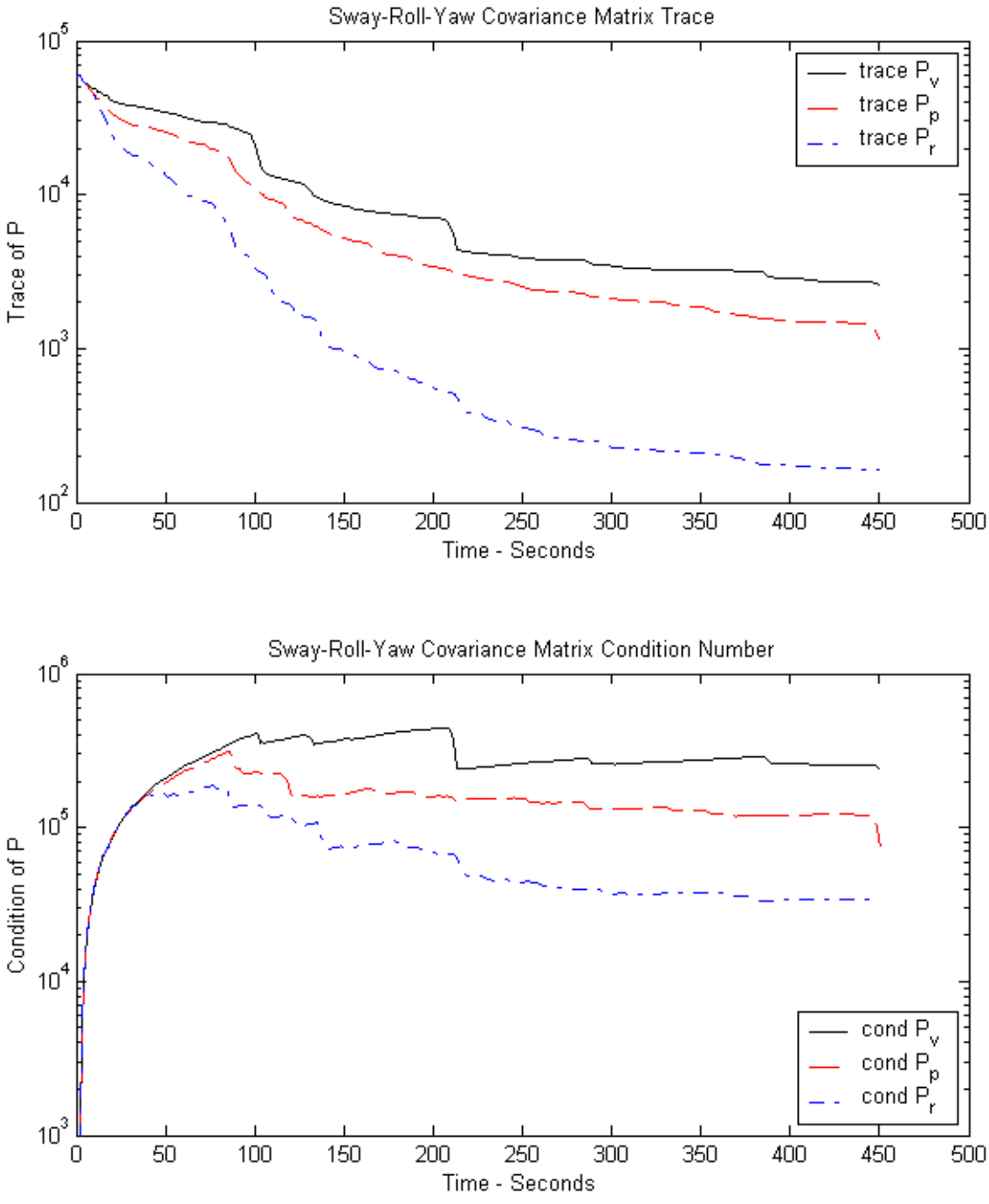


Figure 5.60 Trace and Condition Number of Covariance Matrices Following First Pass of Low–Pass Filtered "4030" Data File

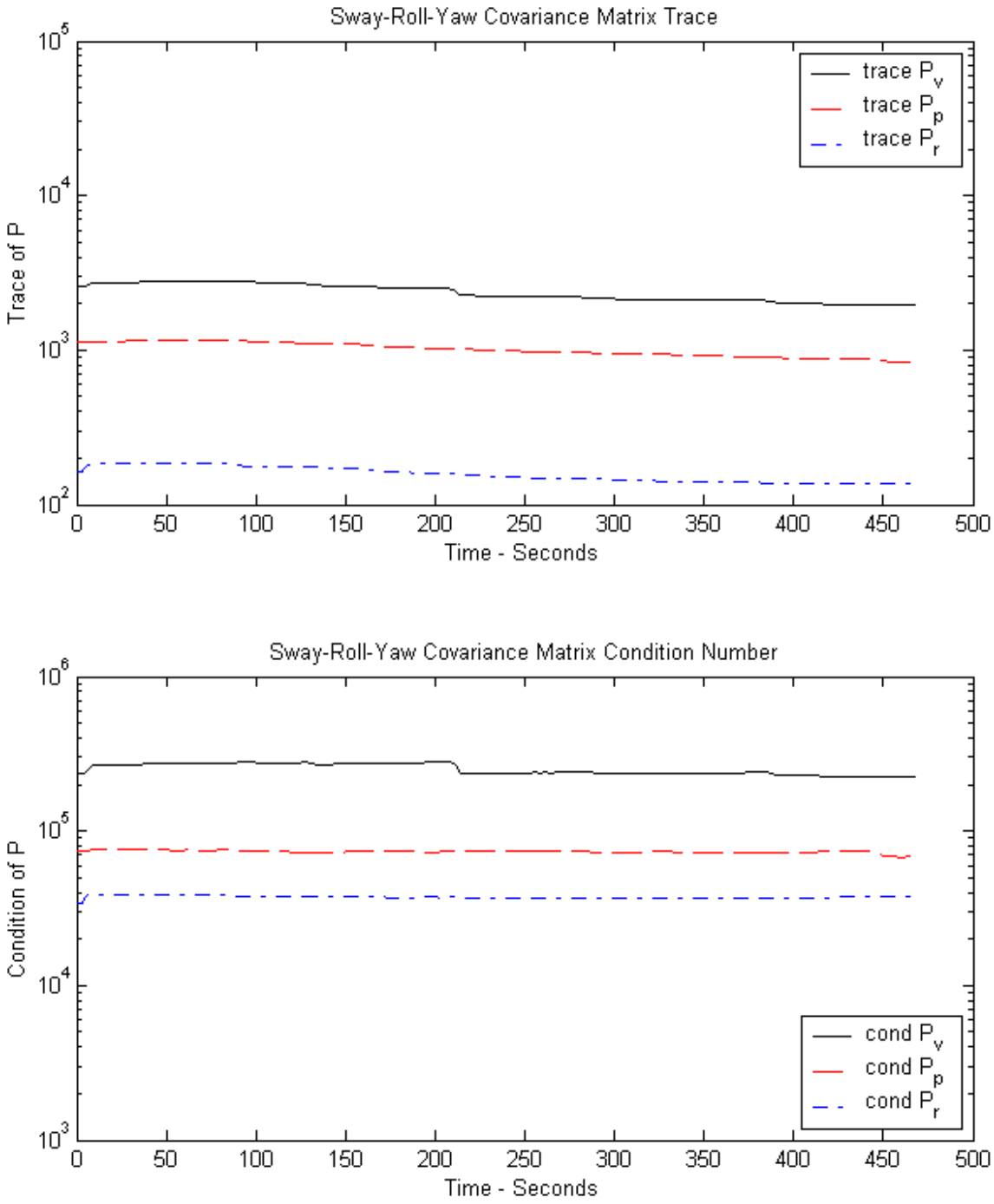
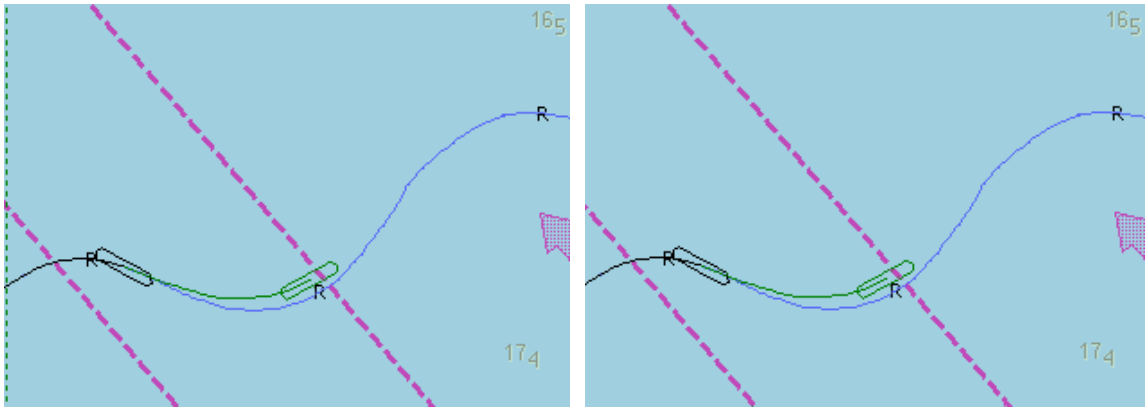
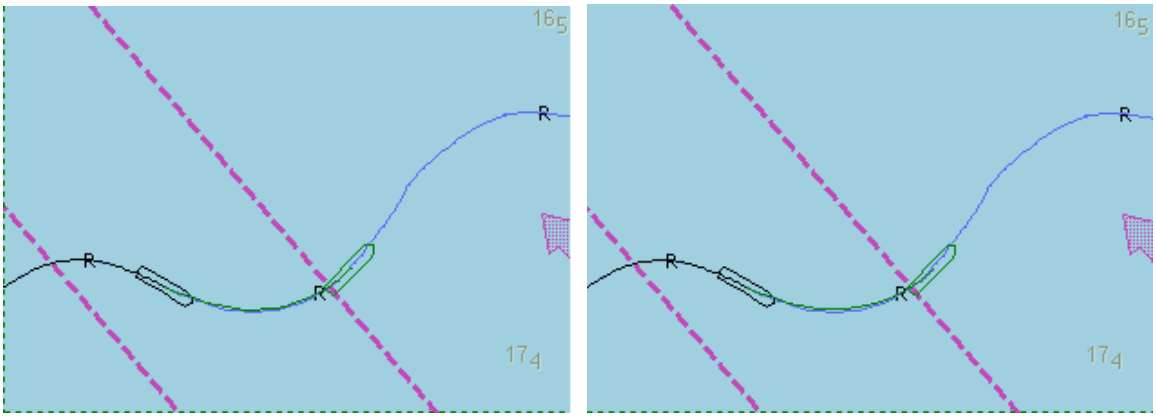


Figure 5.61 Trace and Condition Number of Covariance Matrices Following Second Pass of Low–Pass Filtered "4030" Data File



1st Pass (t = 267 sec) 2nd Pass (t = 267 sec)



1st Pass (t = 282 sec) 2nd Pass (t = 282 sec)

Figure 5.62 USCGC HEALY Zig-Zag Path Prediction for First and Second Pass of Low–Pass Filtered "4030" Data File

b. Surge Parameter Estimation Using Low–Pass Filtered "4030" Data File

Non-linear surge equation parameter identification was also performed while processing the filtered "4030" data file. The states involved are displayed in Figure 5.63. Notice the reduced noise, especially in the yaw rate trace.

Progress of the surge parameter estimation for passes one and two of the filtered "4030" data file is displayed in Figures 5.64 and 5.65, respectively. Parameter convergence is similar to the unfiltered data set.

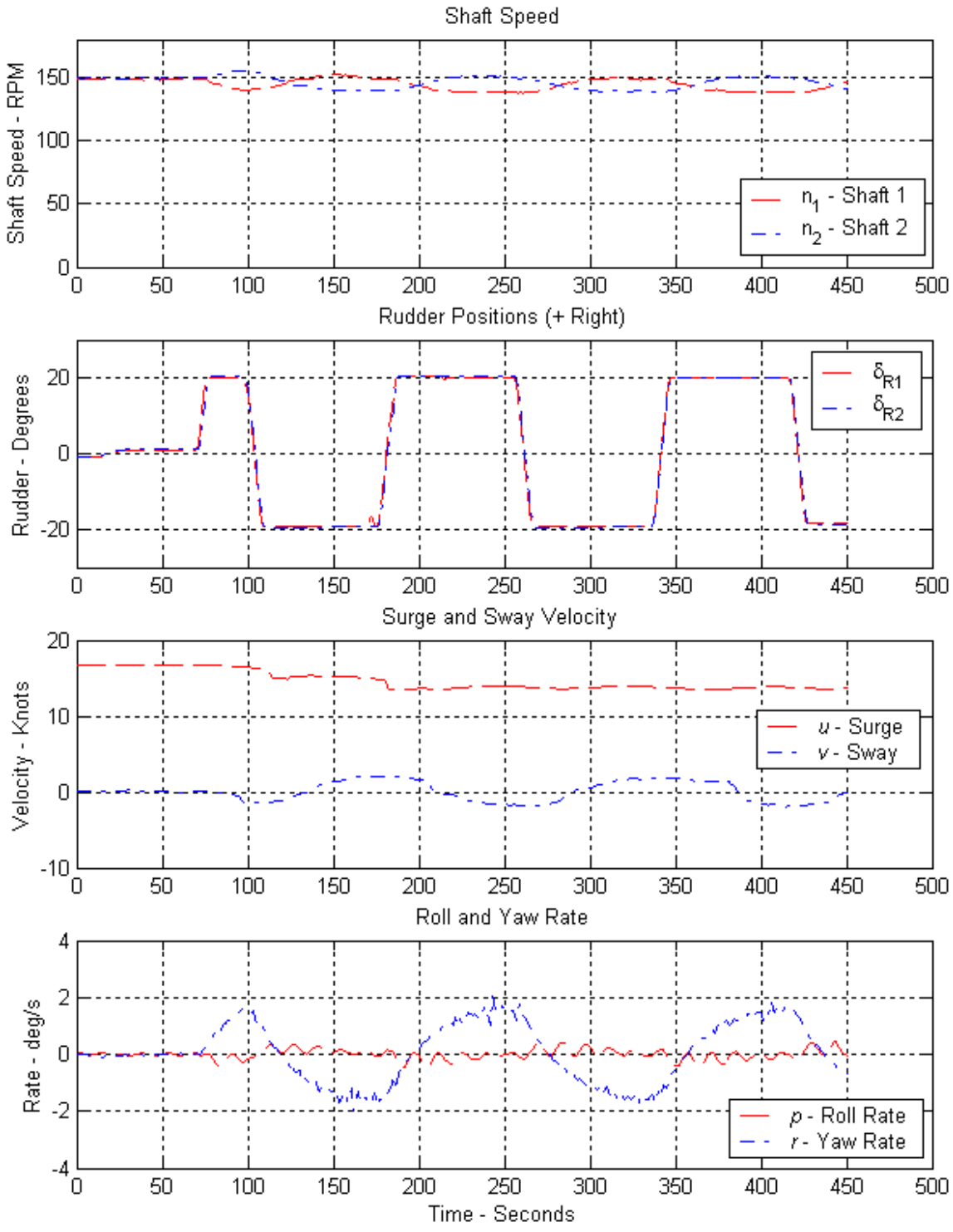


Figure 5.63 Low–Pass Filtered “4030” Vessel States for Surge Parameter Estimation

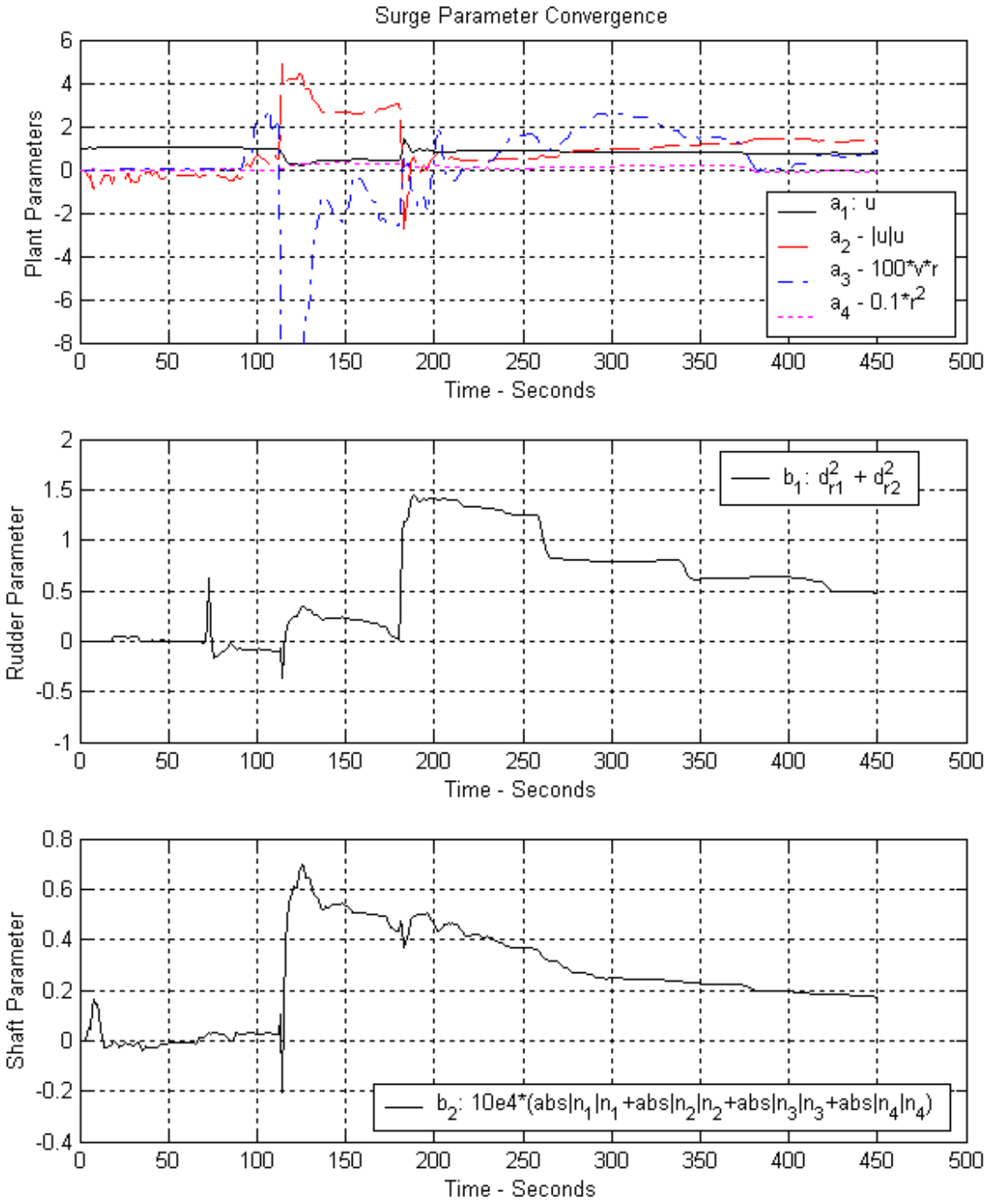


Figure 5.64 Surge Parameter Estimation following First Pass of Low–Pass Filtered "4030" Data File ($\lambda = 0.995$)

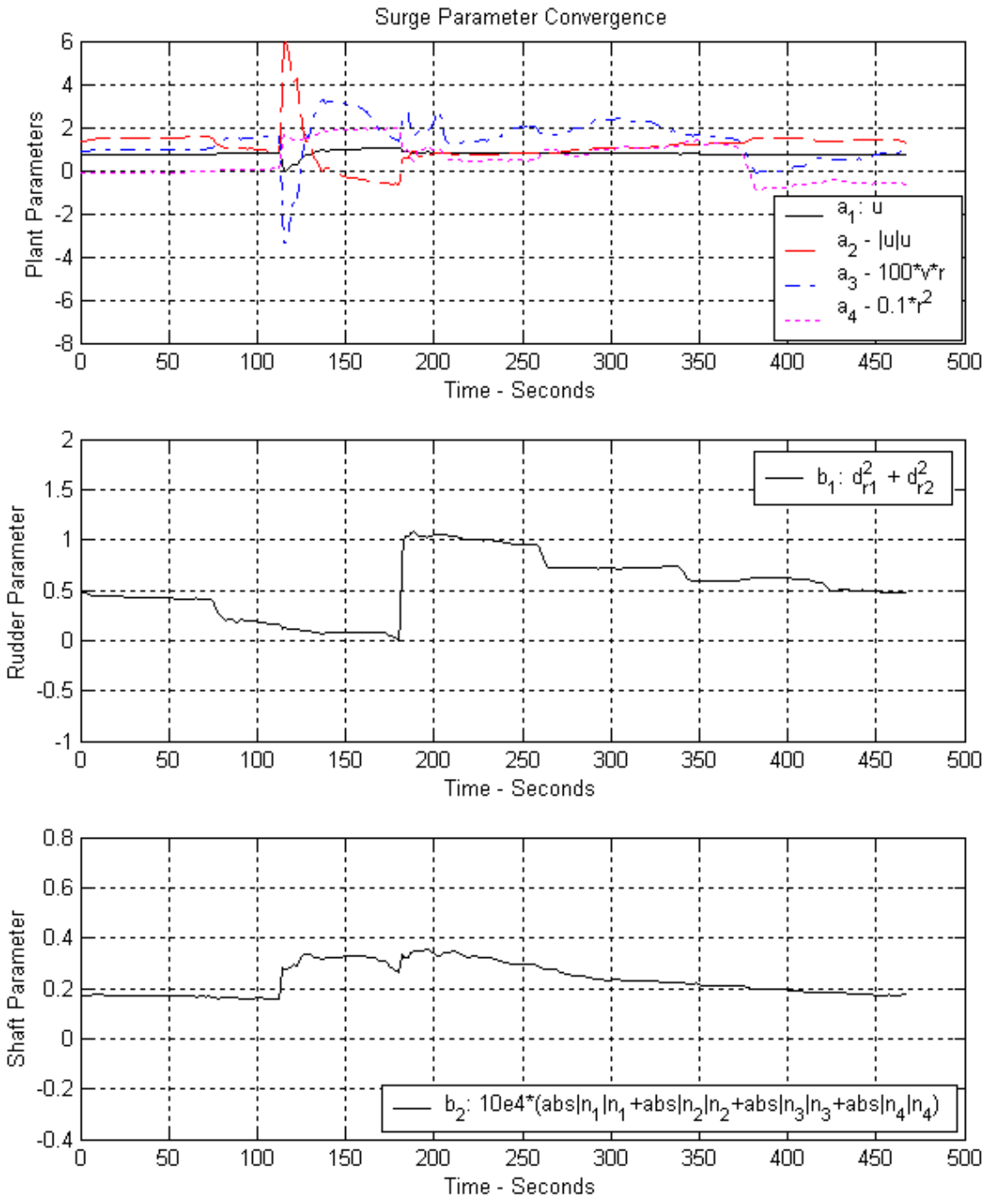


Figure 5.65 Surge Parameter Estimation following Second Pass of Low–Pass Filtered "4030" Data File ($\lambda = 0.995$)

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND RECOMMENDATIONS

A. GENERAL CONCLUSIONS AND RECOMMENDATIONS

Real–time recursive parameter identification was applied to surface vessel modeling for maneuvering path prediction. An end-to-end system was designed to simulate vessel motion, identify parameters and estimate future path. The data were then overlaid in real–time on an electronic chart image taking into account environmental conditions. The system uses extended least–squares (ELS) parameter identification for path prediction rather than implementing a predetermined vessel model. This approach allows the system to be installed on most platforms without prior knowledge of system dynamics provided appropriate vessel states are available. In addition, the system continually tunes to actual environmental conditions, including vessel ballasting, current, wind and sensor biases.

Suitable path prediction performance was established in real–world maneuvering conditions. Sufficient fidelity was demonstrated to recommend that maneuvering path prediction algorithms be incorporated into the Navy's NAVSSI electronic charting system. Figure 6.1 illustrates how maneuvering path prediction improves bridge team situational awareness when compared to the currently implemented instantaneous velocity vector. Vessel states and environmental data are readily available on most large naval vessels. The NAVSSI system should have a module to process the data and provide a real–time depiction of a vessel's future path over the ground.

In addition to path prediction, the sway–roll–yaw model allows prediction of maximum roll angle during maneuvering. Maximum roll angle prediction takes into account environmental conditions and removes the guesswork from selecting the maximum rudder angle available, thus enhancing carrier flight deck safety and improving operational effectiveness.

Insufficient time was available to fully investigate pre-filtering vessel state data prior to parameter estimation. Additional effort should be put into determining optimum noise filtering techniques and cutoff frequencies.

The next development step should be a real–time shipboard demonstration to verify utility to bridge watch standers. Two modifications are required for real–time operation. First, the Data Acquisition Module needs to be modified to accept a serial data stream from a sea trials data logging system. Second, the Track Display Module should be interfaced with a real ECDIS charting and display program. Several commercial off–the–shelf software packages are available that can be driven by LabView through standard Windows interfaces.

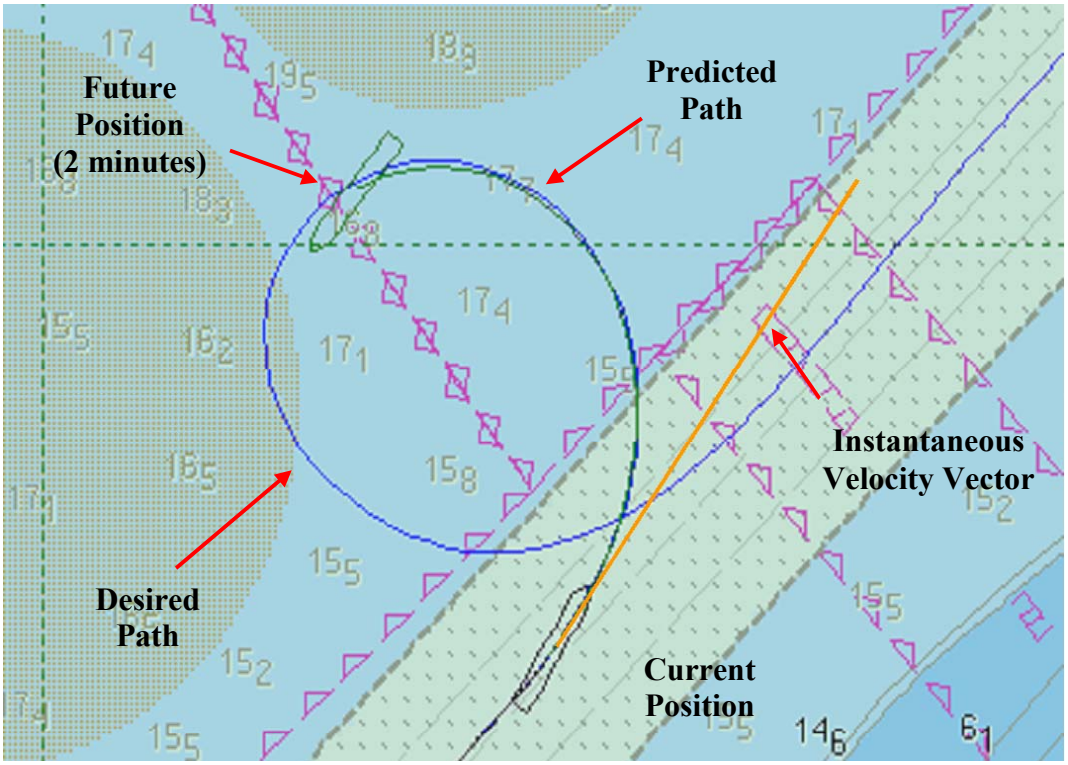


Figure 6.1 Path Prediction vs. Instantaneous Velocity Vector

B. SPECIFIC CONCLUSIONS AND RECOMMENDATIONS

- The sway–roll–yaw and surge models implemented are sufficient for maneuvering at fairly steady ahead bells (power settings). A multiple model approach is required to expand the envelope to include backing, twisting or tug maneuvering.
- Both the sway–yaw and sway–roll–yaw models are suitable for path prediction. The yaw rate only model was useful for predicting steady–state turn rate and radius, but unsuitable for maneuvering situations. It is recommended that the sway–roll–yaw model be used in operational designs so maximum roll angle may be estimated in addition to vessel path.
- The linear sway–roll–yaw parameter estimation model performed quite well. However, there is room for improvement during high bandwidth control applications. Adding non-linear, linear in parameter terms may be warranted.
- The final configuration for the sway–roll–yaw ELS algorithm with conditional updating, directional forgetting and covariance resetting proved quite robust, especially with the real–world, noisy sea trials data. However, additional testing with longer data sets is warranted to verify stability during typical ocean passages.
- Wind disturbance is currently implemented in the bias terms, limiting the validity to small displacements around the average heading. It is recommended that future versions implement a full wind disturbance model. Software hooks exist in the state vector to add a full wind model.
- The USCGC HEALY sea trials data included sideslip velocity (v) from a Doppler transducer. A sideslip state observer is needed on vessels without a sideslip Doppler transducer.
- Further investigation of data pre-processing is warranted. Some noise filtering clearly improves eigenvalue prediction and stability. The optimum technique and cutoff frequency is still to be determined.

- Additional testing is required to evaluate the surge equation model in real–world conditions. The data sets available from the USCGC HEALY sea trials were all captured at nearly constant power settings. The model should be evaluated with throttle transients.
- Real–time performance should be demonstrated during a large vessel sea trial. The Data Acquisition Module will need to be modified to capture and condition real–time serial data from the sea trials instrumentation package.
- Improved ECN functionality is required for a sea trials demonstration. The path prediction module will need to pass data to a commercial ECN module via the built-in LabView X-Windows interface.
- Investigate the optimum display configuration. Consideration should be given to only displaying path prediction during significant maneuvering. In other words, a null zone may need to be implemented to prevent nuisance wandering of the path prediction display during routine course keeping control inputs. The instantaneous velocity vector should be displayed inside the null zone.

APPENDIX A. VESSEL RIGID BODY DYNAMICS

This Appendix amplifies the rigid body dynamics presented in Chapter II. The 6–DOF equations of motion are systematically reduced by eliminating degrees of freedom. Suitability of the reduced systems is discussed with respect to surface vessel navigation.

All vessel models used in this research are derived from the rigid body equations of motion. The 6–DOF equations motion are shown below in vector form [Ref. 1]:

$$M_{RB}\,\dot{v} + C_{RB}(v)\,v = \tau_{RB} \tag{A.1}$$

where M_{RB} is the rigid body inertial matrix, C_{RB} is the Coriolis ($\varpi \times v$) and centripetal ($\varpi \times (\varpi \times r_G)$) matrix, v is the body–fixed linear and angular velocity vector ($[u, v, w, p, q, r]^{\mathrm{T}}$) and τ is the vector of external forces and moments ($[X, Y, Z, K, M, N]^{\mathrm{T}}$).

From Equation A.1, the 6–DOF equations are expanded by terms [Ref. 1]:

$$m\left[\dot{u} - vr + wq - x_G\left(q^2 + r^2\right) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})\right] = X \tag{A.2}$$

$$m\left[\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G\left(qr - \dot{p}\right) + x_G(qp + \dot{r})\right] = Y \tag{A.3}$$

$$m\left[\dot{w} - uq + vp - z_G(p^2 + q^2) + x_G\left(rp - \dot{q}\right) + y_G(rq + \dot{p})\right] = Z \tag{A.4}$$

$$I_x\dot{p} + \left(I_z - I_y\right)qr + m\left[y_G\left(\dot{w} - uq + vp\right) - z_G\left(\dot{v} - wp + ur\right)\right] = K \tag{A.5}$$

$$I_y\dot{q} + \left(I_x - I_z\right)rp + m\left[z_G\left(\dot{u} - vr + wq\right) - x_G\left(\dot{w} - uq + vp\right)\right] = M \tag{A.6}$$

$$I_z\dot{r} + \left(I_y - I_x\right)pq + m\left[x_G\left(\dot{v} - wp + ur\right) - y_G\left(\dot{u} - vr + wq\right)\right] = N \tag{A.7}$$

where *X, Y, Z* are hydrodynamic and external disturbance forces and *K, M* and *N* are hydrodynamic and external disturbance moments.

A full 6–DOF motion model is appropriate in some circumstances, such as modeling the outside visual scene for a bridge simulator. However, simplified models are generally used for surface navigation. In fact, the primary objective of this thesis is to estimate and display future vessel position on an electronic chart display system (ECDIS). By definition, an electronic chart is limited to depicting motion in the *x–y* tangent plane,

making a 2–DOF (*u, r*) or 3–DOF (*u ,v, r*) model appropriate. However, in some cases it is useful to display roll angle information using a 4–DOF (*u ,v, p, r*) model. A good example is the display of maximum aircraft carrier roll angle expected during a maneuver. This could assist the Officer of the Deck (OOD) with restricting rudder movement during flight operations to prevent aircraft from sliding off the flight deck. Other vessels have similar roll angle limitations and prediction of maximum roll angle during maneuvering may prove useful. In general, pitch and heave motions can be ignored for surface navigation.

The 6–DOF equations of motion are reduced to 4–DOF and simplified with the following assumptions:

1) Vessel is symmetric around the *x–z* plane ($I_{xy} = I_{yz} = y_G = 0$).
2) Vessel has a homogeneous mass distribution.
3) The body fixed origin is selected as $r_G = \begin{bmatrix} x_G & 0 & z_G \end{bmatrix}^T$ ($I_{xz} = 0$).
4) Pitch and heave are ignored ($q = w = 0$).

This yields four simplified non-linear equations:

$$m\left(\dot{u} - vr - x_G r^2 + z_G pr\right) = X_{hydro} + X_{ext} \tag{A.8}$$

$$m\left(\dot{v} + ur + x_G \dot{r} - z_G \dot{p}\right) = Y_{hydro} + Y_{ext} \tag{A.9}$$

$$I_z \dot{p} - mz_G\left(\dot{v} + ur\right) = K_{hydro} - W\overline{GM}_T \phi + K_{ext} \tag{A.10}$$

$$I_z \dot{r} + mx_G\left(\dot{v} + ur\right) = N_{hydro} + N_{ext}\,. \tag{A.11}$$

Note: The sway–roll–yaw equations decouple from the surge equation when small perturbations are assumed. Surge velocity (*u*) is replaced by a mean forward speed (U_0).

The 4–DOF state vector, including angles is as follows:

$$x = \begin{bmatrix} u \\ v \\ p \\ \phi \\ r \\ \psi \end{bmatrix} \quad and \quad \dot{x} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{p} \\ p \\ \dot{r} \\ r \end{bmatrix}. \tag{A.12}$$

The 4–DOF equations of motion are further reduced to 3–DOF by eliminating the roll axis ($p = 0$):

$$m\left(\dot{u} - vr - x_G r^2\right) = X_{hydro} + X_{ext} \tag{A.13}$$

$$m\left(\dot{v} + ur + x_G \dot{r}\right) = Y_{hydro} + Y_{ext} \tag{A.14}$$

$$I_z \dot{r} + m x_G \left(\dot{v} + ur\right) = N_{hydro} + N_{ext}. \tag{A.15}$$

The 3–DOF state vector, including angles is as follows:

$$x = \begin{bmatrix} u \\ v \\ r \\ \psi \end{bmatrix} \quad and \quad \dot{x} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \\ r \end{bmatrix}. \tag{A.16}$$

The 3–DOF equations are reduced to 2–DOF by dropping sway motion ($v = 0$):

$$m\left(\dot{u} - x_G r^2\right) = X_{hydro} + X_{ext} \tag{A.17}$$

$$I_z \dot{r} + m x_G \, ur = N_{hydro} + N_{ext}. \tag{A.18}$$

The 2–DOF state vector, including angles is as follows:

$$x = \begin{bmatrix} u \\ r \\ \psi \end{bmatrix} \quad and \quad \dot{x} = \begin{bmatrix} \dot{u} \\ \dot{r} \\ r \end{bmatrix}. \tag{A.19}$$

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. SURFACE VESSEL DYNAMIC MODELS

An overview of linear and non-linear surface vessel models is required to select appropriate plant models for recursive parameter estimation. This appendix provides the background information for the vessel models presented in Chapter II.

All models are based on small perturbation decoupling of the sway–yaw–roll modes from the surge equation. The general goal is to find a model with slowly varying parameters that is valid over a large range of operating conditions. Linear models are preferred; however, recursive parameter estimation can handle non-linear models that are linear in parameters [Ref. 2]. For the interested reader, Fossen [Ref. 1] provides an excellent compilation of dynamic models for both surface and sub-surface vessels. His text provides the foundation for much of the following discussion on vessel modeling.

The models represented are extendable to multi-shaft and multi-rudder vessels. However, simply appending additional inputs to the control vector (e.g.. $\delta_R = \left[\delta_{R1} + \delta_{R2}\right]^T$) does not provide suitable input for parameter estimation. This is due to lack of independent, persistent excitation of the control inputs. In other words, the rudders and shafts are typically operated in tandem. Heuristically, this means the recursive parameter estimator cannot determine which rudder is providing the control power. The method used in this research is to add control inputs assuming linear superposition as follows:

$$|n|\, n = |n_1|\, n_1 + |n_2|\, n_2 + |n_3|\, n_3 + |n_4|\, n_4 \text{ for multi–shaft vessels} \tag{B.1}$$

$$\delta_R = \delta_{R1} + \delta_{R2} \text{ and } \delta_R{}^2 = \delta_{R1}{}^2 + \delta_{R2}{}^2 \text{ for multi–rudder vessels.} \tag{B.2}$$

These simplifications create limitations. For instance, summing shaft RPM squared for a multi–shaft vessel will not allow the parameter estimator to resolve the "twisting" effect of differential engine orders. However, the nice convergence of control parameters more than offsets limitations during infrequent, non-tandem control inputs.

A. STEERING EQUATIONS

This thesis studies the suitability of three linearized steering (sway–roll–yaw) models for vessel path prediction. As discussed earlier, sway–roll–yaw motion is de-coupled from surge using small perturbation theory. Surge velocity, u, is replaced by mean forward speed, U_0. The models are presented in order of increasing complexity.

1. Nomoto 1st Order Steering Equations

The simplest set of steering equations is Nomoto's 1st order model. It comes from the 2–DOF surge and yaw equations of motion. Substituting $u = U_0$ decouples the yaw equation from the surge equation:

$$I_z \dot{r} + m x_G u_0 r = N . \tag{B.3}$$

Rearranging and substituting steering time constant, $T = I_z / m x_G u_0$ and rudder gain $K = N /(\delta_R \cdot m x_G u_0)$, yields the Nomoto's 1st Order steering equation [Ref. 8]:

$$T \dot{r} + r = K \delta_R \tag{B.4}$$

or alternately:

$$T \ddot{\psi} + \dot{\psi} = K \delta_R \text{ with } \dot{\psi} = r . \tag{B.5}$$

Reorganized in state-space form ($\dot{x} = A x + B u$ and $y = C x$), Nomoto's 1st Order model becomes:

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} \psi \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix} \delta_R \text{ (Time constant form)} \tag{B.6}$$

or:

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & a_{22} \end{bmatrix} \begin{bmatrix} \psi \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \end{bmatrix} \delta_R \text{ (Parameter form)} \tag{B.7}$$

where:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \tag{B.8}$$

$$B = \begin{bmatrix} 0 \\ \dfrac{K}{T} \end{bmatrix} \tag{B.9}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{B.10}$$

$$x = \begin{bmatrix} \psi \\ r \end{bmatrix},\ y = \psi \text{, and } u = \delta_R . \tag{B.11}$$

The controllability matrix, $U = [B \ \ AB \ \ldots \ A^{n-1}B]$, is full rank:

$$U = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 0 & \dfrac{K}{T} \\ \dfrac{K}{T} & -\dfrac{K}{T^2} \end{bmatrix}. \tag{B.12}$$

The system is also observable with measured states ψ and δ_R. The observability matrix, $V = [C^* \ A^*C^* \ \ldots \ (A^*)^{n-1}C]$ is full rank:

$$V = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{B.13}$$

Recursive parameter estimation is performed in discrete time. As a result, the zero order hold (ZOH) discrete version of Equation B.7 with sampling interval Δ is taken as:

$$x[k+1] = F\ x[k] + G\ u[k] \tag{B.14}$$

where $F = e^{A\Delta} \approx I + A\,\Delta$, $G = B\,\Delta$, and Δ is sample interval.

Expanding Equation B.14 yields the following discrete version of Nomoto's 1st Order model:

$$\begin{bmatrix} \psi[k+1] \\ r[k+1] \end{bmatrix} = \begin{bmatrix} 1 & \Delta \\ 0 & f_{22} \end{bmatrix} \begin{bmatrix} \psi[k] \\ r[k] \end{bmatrix} + \begin{bmatrix} 0 \\ g_2 \end{bmatrix} \delta_R[k]. \tag{B.15}$$

It is convenient to drop the known heading angle integration term ($\psi[k+1] = \psi[k] + \Delta\, r[k]$) for parameter estimation. This reduces Equation B.15 to a linear, first–order parameter estimation problem:

$$r[k+1] = f\ r[k] + g\ \delta_R[k]. \tag{B.16}$$

Vessel steady–state turn radius (R) is a useful planning parameter. It is approximated with following relationship [Ref. 1]:

$$R \cong \frac{U}{r} \tag{B.17}$$

where $U = \sqrt{u^2 + v^2} \approx u_0$ for the 2-DOF model.

Solving Equation B.6 in steady–state ($\dot{r} = 0$) yields the first–order Nomoto approximation of turn radius:

$$R \cong \frac{u_0}{K\delta_R}. \tag{B.18}$$

The simplicity of the Nomoto 1st Order model results in limitations. On the good side, only heading angle (ψ) and rudder deflection (δ_R) need to be measured to implement the model. However, the order reducing assumptions result in a loss of fidelity in two significant areas:

1) Vessels typically exhibit 2nd order effects due to sway–yaw coupling. However, the model only captures the dominant first–order behavior. As a result, the Nomoto 1st order model will not capture yaw rate overshoots during dynamic maneuvering and should be restricted to small rudder order yaw dynamics [Ref. 8].

2) The Nomoto 1st Order model exhibits slower frequency response than higher order models. The Nomoto's 1st Order time constant, T, is approximately an aggregate of higher order model time constants. Referring to the three time constants in Nomoto's 2nd Order model (presented in section 2), T is approximated by $T = T_1 + T_2 - T_3$ [Ref. 1]. Thus, the model is only suitable for predicting steady–state response and mild, low frequency maneuvering.

2. 2nd Order Steering Equations

A second–order sway–yaw model is obtained from the 3–DOF equations of motion. The sway–yaw equations are decoupled from surge by assuming small perturbations around a mean forward speed ($u = U_0$):

$$m\left(\dot{v}+u_0 r+x_G \dot{r}\right)=Y_{hydro}+Y_{ext} \tag{B.19}$$

$$I_z \dot{r}+m x_G\left(\dot{v}+u_0 r\right)=N_{hydro}+N_{ext}\,. \tag{B.20}$$

Placing the body axis origin at the center of gravity ($x_G = 0$) and substituting the first–order Taylor series expansion hydrodynamic terms yields the following sway–yaw equations [Ref. 8]:

$$m\left(\dot{v}+u_0 r\right)=Y_v v+Y_{\dot{v}}\dot{v}+Y_r r+Y_{\dot{r}}\dot{r}+Y_{\delta R}\delta_R \tag{B.21}$$

$$I_z \dot{r}=N_v v+N_{\dot{v}}\dot{v}+N_r r+N_{\dot{r}}\dot{r}+N_{\delta R}\delta_R\,. \tag{B.22}$$

Placing Equations B.21 and B.22 in state space form $M\dot{v}+N(u_0)v=b\,\delta_R$ yields the Nomoto model:

$$\underbrace{\begin{bmatrix} m-Y_{\dot{v}} & -Y_{\dot{r}} & 0 \\ -N_{\dot{v}} & I_z-N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{M}\begin{bmatrix}\dot{v}\\ \dot{r}\\ \dot{\psi}\end{bmatrix}+\underbrace{\begin{bmatrix} -Y_v & m u_0-Y_r & 0 \\ -N_v & -N_r & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{N}\begin{bmatrix} v\\ r\\ \psi\end{bmatrix}=\underbrace{\begin{bmatrix} Y_\delta \\ N_\delta \\ 0\end{bmatrix}}_{b}\delta_R \tag{B.23}$$

The terms of Equation B.23 may be rearranged in the more familiar $\dot{x}=A\,x+B\,u$ and $y=C\,x$ format:

$$\begin{bmatrix}\dot{v}\\ \dot{r}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{21} & 0 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} v\\ r\\ \psi\end{bmatrix}+\begin{bmatrix} b_1\\ b_2\\ 0\end{bmatrix}\delta_R \tag{B.24}$$

where:

$$A=-M^{-1}N=\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{B.25}$$

$$B=M^{-1}b=\begin{bmatrix} b_1\\ b_2\\ 0\end{bmatrix} \tag{B.26}$$

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{B.27}$$

$$\delta_R = \delta_{R1} + \delta_{R2} \text{ for multiple rudder vessels} \tag{B.28}$$

with the coefficients:

$$a_{11} = \frac{\left(I_z - N_{\dot{r}}\right) Y_v + Y_{\dot{r}} N_v}{\det\left(M\right)} \tag{B.29}$$

$$a_{12} = \frac{\left(I_z - N_{\dot{r}}\right)\left(Y_r - mu_0\right) + Y_{\dot{r}} N_r}{\det\left(M\right)} \tag{B.30}$$

$$a_{21} = \frac{\left(m - Y_{\dot{v}}\right) N_v + N_{\dot{v}} Y_v}{\det\left(M\right)} \tag{B.31}$$

$$a_{22} = \frac{\left(m - Y_{\dot{v}}\right) N_r + N_{\dot{v}}\left(Y_r - mu_0\right)}{\det\left(M\right)} \tag{B.32}$$

$$b_1 = \frac{\left(I_z - N_{\dot{r}}\right) Y_\delta + Y_{\dot{r}} N_\delta}{\det\left(M\right)} \tag{B.33}$$

$$b_2 = \frac{\left(m - Y_{\dot{v}}\right) N_\delta + N_{\dot{v}} Y_\delta}{\det\left(M\right)} \tag{B.34}$$

$$\det\left(M\right) = \left(m - Y_{\dot{v}}\right)\left(I_z - N_{\dot{r}}\right) - N_{\dot{v}} Y_{\dot{r}} \,. \tag{B.35}$$

A version of this system without the body axis restricted to the center of gravity ($x_G \neq 0$), is presented by Davidson and Schiff (1946) [Ref. 1]:

$$\underbrace{\begin{bmatrix} m - Y_{\dot{v}} & mx_G - Y_{\dot{r}} & 0 \\ mx_G - N_{\dot{v}} & I_z - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{M} \begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} + \underbrace{\begin{bmatrix} -Y_v & mu_0 - Y_r & 0 \\ -N_v & mx_G u_0 - N_r & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{N} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} = \underbrace{\begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix}}_{b} \delta_R \,. \tag{B.36}$$

This model reduces to the same state space structure as Equation B.24. However, the individual coefficients are as follows:

$$a_{11} = \frac{\left(I_z - N_{\dot{r}}\right) Y_v - \left(mx_G - Y_{\dot{r}}\right) N_v}{\det\left(M\right)} \tag{B.37}$$

$$a_{12} = \frac{\left(I_z - N_{\dot{r}}\right)\left(Y_r - mu_0\right) - \left(mx_G - Y_{\dot{r}}\right)\left(N_r - mx_G u_0\right)}{\det\left(M\right)} \tag{B.38}$$

$$a_{21} = \frac{\left(m - Y_{\dot{v}}\right)N_v - \left(mx_G - N_{\dot{v}}\right)Y_v}{\det\left(M\right)} \tag{B.39}$$

$$a_{22} = \frac{\left(m - Y_{\dot{v}}\right)\left(N_r - mx_G u_0\right) - \left(mx_G - N_{\dot{v}}\right)\left(Y_r - mu_0\right)}{\det\left(M\right)} \tag{B.40}$$

$$b_1 = \frac{\left(I_z - N_{\dot{r}}\right)Y_\delta - \left(mx_G - Y_{\dot{r}}\right)N_\delta}{\det\left(M\right)} \tag{B.41}$$

$$b_2 = \frac{\left(m - Y_{\dot{v}}\right)N_\delta - \left(mx_G - N_{\dot{v}}\right)Y_\delta}{\det\left(M\right)} \tag{B.42}$$

$$\det\left(M\right) = \left(m - Y_{\dot{v}}\right)\left(I_z - N_{\dot{r}}\right) - \left(mx_G - N_{\dot{v}}\right)\left(mx_G - Y_{\dot{r}}\right). \tag{B.43}$$

All parameters are observable and may be estimated provided the sway and yaw time constants are different [Ref. 1].

Both the Nomoto and Davison/Schiff coupled sway–yaw state space models reduce to the following linear system in discrete time:

$$\begin{bmatrix} v[k+1] \\ r[k+1] \\ \psi[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & 0 \\ f_{21} & f_{21} & 0 \\ 0 & \Delta & 1 \end{bmatrix} \begin{bmatrix} v[k] \\ r[k] \\ \psi[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ 0 \end{bmatrix} \delta_R[k]. \tag{B.44}$$

As with the first–order steering equation, it is convenient to drop heading angle integration ($\psi[k+1] = \psi[k] + \Delta\, r[k]$) for parameter estimation leaving the following discrete model:

$$\begin{bmatrix} v[k+1] \\ r[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} v[k] \\ r[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \delta_R[k]. \tag{B.45}$$

Note that yaw angle (ψ) may be dropped due to a lack of coupling between yaw angle and yaw rate.

Many ships do not have Doppler transducers for the accurate measurement of sway velocity (v). In 1957, Nomoto proposed a modified second–order steering equation

by dropping sway velocity. This yields a yaw rate to rudder order transfer function of [Ref. 1]:

$$\frac{r}{\delta_R} = \frac{K_R(1+T_3 s)}{(1+T_1 s)(1+T_2 s)} \quad \text{or} \quad T_1 T_2\, \ddot{r} + (T_1 + T_2)\,\dot{r} + r = K_R(\delta_R + T_3 \dot{\delta}_R). \tag{B.46}$$

Substituting $r = \dot{\psi}$ into Equation B.46 yields the heading equation:

$$T_1 T_2\, \dddot{\psi} + (T_1 + T_2)\,\ddot{\psi} + \dot{\psi} = K_R(\delta_R + T_3 \dot{\delta}_R) \tag{B.47}$$

where the time constants and rudder gain are taken from the Davidson and Schiff model elements (Equation B.36):

$$T_1 T_2 = \frac{\det(M)}{\det(N)} \tag{B.48}$$

$$(T_1 + T_2) = \frac{n_{11} m_{22} + n_{22} m_{11} - n_{12} m_{21} - n_{21} m_{12}}{\det(N)} \tag{B.49}$$

$$K_R = \frac{n_{21} b_1 - n_{11} b_2}{\det(N)} \tag{B.50}$$

$$K_R T_3 = \frac{m_{21} b_1 - m_{11} b_2}{\det(N)}\,. \tag{B.51}$$

Placing Equation B.47 in $\dot{x} = A\,x + B\,u$ state space form yields:

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -\dfrac{(T_1 + T_2)}{T_1 T_2} \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ K_R & K_R T_3 \end{bmatrix} \begin{bmatrix} \delta_R \\ \dot{\delta}_R \end{bmatrix}. \tag{B.52}$$

In discrete time, using the ZOH approximation, Nomoto's 2nd Order model becomes:

$$\begin{bmatrix} r[k+1] \\ \dot{r}[k+1] \end{bmatrix} = \begin{bmatrix} 1+\Delta & 0 \\ \Delta & f_{22} \end{bmatrix} \begin{bmatrix} r[k] \\ \dot{r}[k] \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} \delta_R[k] \\ \dot{\delta}_R[k] \end{bmatrix}. \tag{B.53}$$

Nomoto's 2nd Order model captures dynamic overshoot in yaw rate and demonstrates improved frequency response when compared to the Nomoto's 1st Order model. However, accurate measurement of the states could prove problematic for parameter es-

timation. A state observer is required to generate $\ddot{\psi}$ from heading measurement (ψ). Typically, double differentiation produces undesirable noise. One solution would be to capture $\dot{\psi}$ and $\ddot{\psi}$ from an inertial measuring unit (IMU). IMUs are an integral component of the Ship's Inertial Navigation System (SINS) installed on all large US Navy vessels.

3. Coupled Sway–Roll–Yaw Steering Equations

A coupled sway–roll–yaw model is obtained from the 4–DOF equations of motion. The sway, roll and yaw equations are decoupled from surge by assuming small perturbations around a mean forward speed ($u = U_{0)}$:

$$m\left(\dot{v} + u_0 r + x_G \dot{r} - z_G \dot{p}\right) = Y_{hydro} + Y_{ext} \tag{B.54}$$

$$I_z \dot{p} - m z_G \left(\dot{v} + u_0 r\right) = K_{hydro} - W\overline{GM}_T \phi + K_{ext} \tag{B.55}$$

$$I_z \dot{r} + m x_G \left(\dot{v} + u_0 r\right) = N_{hydro} + N_{ext} \tag{B.56}$$

Christensen and Blanke proposed the following non-linear model based on Equations B.54 through B.56 [Ref. 1]:

$$\begin{bmatrix} m - Y_{\dot{v}} & -mz_G - Y_{\dot{p}} & mx_G - Y_{\dot{r}} & 0 & 0 \\ -mz_G - K_{\dot{v}} & I_x - K_{\dot{p}} & 0 & 0 & 0 \\ mx_G - N_{\dot{v}} & 0 & I_z - N_{\dot{r}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \tag{B.57}$$

$$\begin{bmatrix} Y_{uv}\lfloor u \rfloor & Y_{\lfloor u \rfloor p}\lfloor u \rfloor & -mu + Y_{ur}u & Y_{uu\phi}u^2 & 0 \\ K_{\lfloor u \rfloor v}\lfloor u \rfloor & K_{up}u + K_p & K_{ur}u & W\overline{GM}_T + K_{uu\phi}u^2 & 0 \\ N_{uv}u & 0 & N_{\lfloor u \rfloor r}\lfloor u \rfloor - mx_G u & N_{\lfloor u \rfloor u\phi}\lfloor u \rfloor u & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} Y_{ext} \\ K_{ext} \\ N_{ext} \\ 0 \\ 0 \end{bmatrix}.$$

Linearizing the model around a mean forward speed (U_0), yields the following state space model structure where $x = [v\ r\ p\ \phi\ \psi]$:

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{p} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ p \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \\ 0 \end{bmatrix} \delta_R \,. \tag{B.58}$$

If the state vector is rearranged to $x = [v\ r\ \psi\ p\ \phi]$, the model takes the alternate form:

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} & a_{15} \\ a_{21} & a_{22} & 0 & a_{24} & a_{25} \\ 0 & 1 & 0 & 0 & 0 \\ a_{41} & a_{42} & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ b_4 \\ 0 \end{bmatrix} \delta_R \,. \tag{B.59}$$

Notice that Equation B.59 may be represented in block diagonal form as:

$$\begin{bmatrix} \dot{x}_\psi \\ \dot{x}_\phi \end{bmatrix} = \begin{bmatrix} A_{\psi\psi} & A_{\psi\phi} \\ A_{\phi\psi} & A_{\phi\phi} \end{bmatrix} \begin{bmatrix} x_\psi \\ x_\phi \end{bmatrix} + \begin{bmatrix} b_\psi \\ b_\phi \end{bmatrix} \delta_R \,. \tag{B.60}$$

This representation is convenient for illustrating the roll coupling effects contained in $A_{\phi\psi}$ and $A_{\psi\phi}$. The previously discussed linear sway–yaw model is simply the dynamics contained in the upper left block ($A_{\psi\psi}$).

The state space model in Equation B.59 is reduced for parameter estimation by eliminating heading integration ($\dot{\psi} = r$):

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix} \delta_R \,. \tag{B.61}$$

Roll angle must be kept as a parameter. This is due to the coupling of roll angle to roll moment. In the sway–yaw model, yaw angle (ψ) was dropped due to lack of coupling between yaw rate and yaw angle.

The discrete–time state space representation of Equation B.61 is:

$$\begin{bmatrix} v[k+1] \\ r[k+1] \\ p[k+1] \\ \phi[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ 0 & 0 & \Delta & 1 \end{bmatrix} \begin{bmatrix} v[k] \\ r[k] \\ p[k] \\ \phi[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ 0 \end{bmatrix} \delta_R[k]. \tag{B.62}$$

Dropping roll angle integration ($\phi[k+1] = \phi[k] + \Delta\, p[k]$), the discrete–time system used for parameter estimation becomes:

$$\begin{bmatrix} v[k+1] \\ p[k+1] \\ r[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \end{bmatrix} \begin{bmatrix} v[k] \\ p[k] \\ r[k] \\ \phi[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \delta_R[k]. \tag{B.63}$$

Notice that the plant matrix is no longer square. This is easily accommodated in the RLS and ELS algorithms. The states are also reordered to correspond to the state vector order, [*u v p r* ϕ ψ …], used in the Data Acquisition and Parameter Estimation modules.

Neglecting roll angle influence on the sway–yaw equations can further reduce the system. Thus, the minimum parameter, sway–roll–yaw model is represented below:

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix} \delta_R. \tag{B.64}$$

The discrete–time the model with states reordered to correspond to the Data Acquisition and Parameter Estimation module state vector, [*u v p r* ϕ ψ …], becomes:

$$\begin{bmatrix} v[k+1] \\ p[k+1] \\ r[k+1] \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & 0 \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & 0 \end{bmatrix} \begin{bmatrix} v[k] \\ p[k] \\ r[k] \\ \phi[k] \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \delta_R[k]. \tag{B.65}$$

B. THE SPEED EQUATION

A single non-linear equation is used to model vessel surge motion. Linear surge models are available, but they are only valid for small ranges around the trimmed vessel speed. As a result, multiple linear trim points are required to model a typical vessel speed range. Assuming *u, v* and *r* are observable vessel states; it is relatively straightforward to develop a non-linear speed equation that is linear in parameters.

The 3–DOF surge equation, A.13, is the starting point for this study. Pitch, roll and heave motions are ignored, leaving the x–direction hydrodynamic forces a function of *u, v, r,* δ_R and Thrust:

$$m\left(\dot{u} - vr - x_G r^2\right) = X_{hydro}\left(u, v, r, \delta_R, T\right) + X_{ext}\,. \tag{B.66}$$

The non-linear model of Blanke (1981) suggests the following parameterization [Ref. 1]:

$$\left(m - X_{\dot{u}}\right)\dot{u} = X_{|u|u}|u|u + \left(1+t\right)T + T_{loss} \tag{B.67}$$

$$T_{loss} = \left(m + X_{vr}\right)vr + \left(mx_G + X_{rr}\right)r^2 + X_{\delta\delta}\ \delta_R{}^2 + X_{ext}\,. \tag{B.68}$$

For the purposes of parameter estimation, the thrust reduction term (*t*) is considered constant around the cruise power setting and propeller thrust is assumed to be proportional to shaft RPM squared in steady–state cruise. Dividing both sides by the mass and added mass term ($m + X_a$) yields the following approximated, non-linear surge equation that is linear in parameters:

$$\dot{u} = a_1\ |u|u + a_2\ vr + a_3\ r^2 + b_1\ \delta_R{}^2 + b_2\ |n|n \tag{B.69}$$

where:

$$\text{Thrust} \propto n^2 \text{ and } n \cong \text{Trim RPM} \tag{B.70}$$

$$|n|\,n = |n_1|\,n_1 + |n_2|\,n_2 + |n_3|\,n_3 + |n_4|\,n_4 \text{ for multi shaft vessels} \tag{B.71}$$

$$\delta_R{}^2 = \delta_{R1}{}^2 + \delta_{R2}{}^2 \text{ for multi rudder vessels.} \tag{B.72}$$

Recursive parameter estimation is performed in discrete–time using the ZOH approximation of Equation B.69:

$$\begin{aligned} u[k+1] = f_1\, u[k] + f_2 \left|u[k]\right| u[k] + f_3\, v[k] r[k] \ldots \\ + f_4\, r[k]^2 + g_1\, \delta_R[k]^2 + g_2 \left|n[k]\right| n[k] \cdot \end{aligned} \tag{B.73}$$

Note the addition of $f_1\, u[k]$ to create a discrete–time equation in the form:

$$u[k+1] = f_1\, u[k] + \textit{Thrust Terms} + \textit{Drag Terms}\,. \tag{B.74}$$

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. NON-DIMENSIONAL PARAMETERIZATION

This appendix describes the Prime system for obtaining non-dimensional vessel coefficients. It is useful to normalize model parameters with respect to vessel speed. This creates constant coefficients over a wide range of operating conditions. Several systems of normalization exist, but the most common is the Prime system. Table C.1 displays the normalization parameters for the Prime – I ($'$) and Prime – II ($''$) systems [Ref. 1]. The length, L, is equal to the distance between the fore and aft perpendiculars and U is the instantaneous speed. The major difference between the Prime – I and Prime – II systems is the use of a reference area ($L*T$) in the Prime – II system. This is analogous to normalization by wing area in aerodynamics. For consistency, all normalization in this thesis uses the Prime – I system.

Unit	**Prime System I**	**Prime System II**
Length	L	L
Mass	$\rho L^3/2$	$\rho L^2 T/2$
Inertia Moment	$\rho L^5/2$	$\rho L^4 T/2$
Time	L/U	L/U
Reference Area	L^2	LT
Position	L	L
Angle	1	1
Linear Velocity	U	U
Angular Velocity	U/L	U/L
Linear Acceleration	U^2/L	U^2/L
Angular Acceleration	U^2/L^2	U^2/L^2
Force	$\rho U^2 L^2/2$	$\rho U^2 LT/2$
Moment	$\rho U^2 L^3/2$	$\rho U^2 L^2 T/2$

Table C.1 Normalization Parameters for the Prime Systems [From Ref. 1]

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. DISCRETE SYSTEM INTEGRATION TECHNIQUES

This appendix describes numerical integration techniques that range in fidelity and computational cost. Numerical integration is required to obtain discrete–time samples from continuous–time math models. Table D.1 summarizes three popular algorithms and the corresponding truncation error. The Data Acquisition module uses the Runge–Kutta 2nd Order algorithm to generate simulated vessel data. Note that surface vessels typically display light damping in the roll mode. Any open–loop integration of models with a roll mode should use the Runge–Kutta 2nd Order or higher techniques. Fossen [Ref. 1] covers each method and corresponding stability region in detail.

Technique	**Method**	**Truncation Error**	**Comments**
Forward Euler	$x[k+1] = x[k] + h\ f(x(k), u(k))$	$O(h)$	Poor results for lightly damped systems. Lowest cost.
Runge–Kutta 2nd Order	$k_1 = f(x(k), u(k))$ $k_2 = f(x(k) + hk_1, u(k))$ $x[k+1] = x[k] + \frac{h}{2}(k_1 + k_2)$	$O(h^2)$	Improved stability. Moderate cost.
Runge–Kutta 4th Order	$k_1 = h\ f(x(k), u(k))$ $k_2 = h\ f(x(k) + k_1/2, u(k))$ $k_3 = h\ f(x(k) + k_2/2, u(k))$ $k_4 = h\ f(x(k) + k_3/2, u(k))$ $x[k+1] = x[k] + \ldots$ $\ldots \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$	$O(h^4)$	Large region of stability. Will work with lightly damped systems. High cost.

Table D.1 Discrete Integration Techniques [From Ref. 1]

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. USCGC HEALY DATA PARAMETERS

Table E.1 lists the data parameters, source, units and resolution available from the USCGC HEALY Performance and Special Sea Trials Data set.

PARAMETER	*SOURCE*	*UNITS*	*RESOLUTION*
HEADING	SHIP GYRO	DEG	0.1
ROLL	SHIP MMS	DEG	0.01
PITCH	SHIP MMS	DEG	0.01
HEAVE	SHIP MMS	CM	1
HEAVE FLAG	SHIP MMS	BOOLEAN	1
ACCELERATION - HORIZONTAL	SHIP MMS	CM/S^2	3.83
ACCELERATION - VERTICAL	SHIP MMS	CM/S^2	0.0625
REL WIND SPEED	SHIP SENSOR	KNOTS	0.1
REL WIND DIRECTION	SHIP SENSOR	DEG	0.1
RUDDER - 1	SHIP SENSOR	DEG	0.1
RUDDER - 2	SHIP SENSOR	DEG	0.1
VOLTAGE STANDARD	NSWCCD BATTERY	V	0.001
WATER SPEED - FORE/AFT	DOPPLER SPEED LOG	KNOTS	0.01
WATER SPEED - PORT/STBD	DOPPLER SPEED LOG	KNOTS	0.01
WATER SPEED FLAG	DOPPLER SPEED LOG	BOOLEAN	1
BOTTOM SPEED - FORE/AFT	DOPPLER SPEED LOG	KNOTS	0.01
BOTTOM SPEED - PORT/STBD	DOPPLER SPEED LOG	KNOTS	0.01
BOTTOM SPEED FLAG	DOPPLER SPEED LOG	BOOLEAN	1
DEPTH	DOPPLER SPEED LOG	M	0.01
DEPTH FLAG	DOPPLER SPEED LOG	BOOLEAN	1
TIME	SHIP DGPS	S	0.1
LATITUDE	SHIP DGPS	S	0.01
LONGITUDE	SHIP DGPS	S	0.01
HEADING	SHIP DGPS	DEG	0.1
ROLL	SHIP DGPS	DEG	0.1
PITCH	SHIP DGPS	DEG	0.1
ATTITUDE FLAG	SHIP DGPS	BOOLEAN	1
DIFFERENTIAL FLAG	SHIP DGPS	BOOLEAN	1
HDOP	SHIP DGPS	NA	1
TIME	NA	S	1
SHAFT SPEED - 1	NSWCCD COUNTER	COUNTS	1
SHAFT SPEED - 2	NSWCCD COUNTER	COUNTS	1
SHAFT POWER - 1	CALCULATED	HP	1
SHAFT POWER - 2	CALCULATED	HP	1
FUEL CONSUMPTION - GEN 1	CALCULATED	GPM	1
FUEL CONSUMPTION - GEN 2	CALCULATED	GPM	1
FUEL CONSUMPTION - GEN 3	CALCULATED	GPM	1
FUEL CONSUMPTION - GEN 4	CALCULATED	GPM	1
FUEL CONSUMPTION - BOILER 2	CALCULATED	GPM	1
TRUE WIND SPEED	CALCULATED	KNOTS	1
TRUE WIND DIRECTION	CALCULATED	DEG	1
X POSITION	CALCULATED	YD	1
Y POSITION	CALCULATED	YD	1

Table E.1 USCGC HEALY (WAGB 20) Data Parameters [From Ref. 5].

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. T. I. Fossen, *Guidance and Control of Ocean Vehicle*s, John Wiley and Sons, New York, 1994.

2. K. J. Åström and B. Wittenmark, *Adaptive Control*, Addison-Wesley Publishing, New York, 1995.

3. L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

4. L. Ljung and T. Söderström, *Theory and Practice of Recursive Parameter Identification*, MIT Press, Cambridge, Massachusetts, 1983.

5. L. Hundley, G. Brodie and S. Intolubbe, *USCGC HEALY (WAGB 20) Performance and Special Sea Trials*, Naval Surface Warfare Center, Carderock Division, 1999.

6. U.S. Coast Guard World Wide Web Homepage, http://www.uscg.mil/images/graphics/drawings/draw.html, August 2003.

7. USCGC HEALY *(WAGB 20)* World Wide Web Homepage, http://www.uscg.mil/pacarea/healy/, August 2003.

8. Tzeng, C., and Chen, J., "Fundamental Properties of Linear Ship Steering Dynamic Models," *Journal of Marine Science and Technology,* Vol. 7, No. 2, pp 79–88, 1999.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Stephen J. Pollard
   Naval Postgraduate School
   Monterey, California

4. Roberto Cristi
   Naval Postgraduate School
   Monterey, California

5. Fotis A. Papoulias
   Naval Postgraduate School
   Monterey, California

6. John P. Powers
   Naval Postgraduate School
   Monterey, California

7. George Brodie
   Naval Surface Warfare Center, Carderock Division
   David Taylor Model Basin
   Bethesda, Maryland

8. NAVSSI Program Office
   Marine Navigation Division (D32)
   Space and Naval Warfare Systems Center (SSC)
   San Diego, California